Title

Evaluation of Bus and Truck Automation Operations Concepts

Authors
Tsao, H. S. Jacob
Zhang, Lan
Lin, Lin
et al.

Publication Date
2004-11-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Evaluation of Bus and Truck Automation Operations Concepts

H.-S. Jacob Tsao, Lan Zhang, Lin Lin, Deepa Batni
San José State University

California PATH Research Report
UCB-ITS-PRR-2004-45

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

Report for Task Order 4236

November 2004

ISSN 1055-1425

TASK ORDER 4236 FINAL REPORT

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

H.-S. Jacob Tsao, Lan Zhang, Lin Lin and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA
San Jose State University

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

TABLE OF CONTENTS

PART III - EVALUATION OF URBAN BUS AUTOMATION: DISTURBANCE TO SURROUNDING TRAFFIC

PART IV: EVALUATION OF INTER-CITY TRUCK AUTOMATION

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

PART I – PROJECT OVERVIEW

PART I: PROJECT OVERVIEW

TABLE OF CONTENTS

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

H.-S. Jacob Tsao, Lan Zhang, Lin Lin and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA
San Jose State University

ABSTRACT

Traffic congestion will continue to worsen and likely worsen at a faster rate than ever. People throughput and freight throughput have become critical issues for California and the rest of the nation. PATH has funded with approximately $125K a research project entitled “Evaluation of Bus and Truck Automation Scenarios” jointly proposed by Jan Botha (Principal Investigator) of Department of Civil and Environmental Engineering and Jacob Tsao (Co-PI) of Department of Industrial and Systems Engineering at San Jose State University. This report summarizes the major findings of the research conducted by Professor Tsao and his assistants with approximately $44K out of the overall funding of $125K for the project; the infrastructure and other aspects of the research are reported by Jan Botha separately.

During the one-year project, we developed detailed operating scenarios for both urban bus automation and inter-city truck automation as well as operating scenarios for conventional alternatives. We also compared the automation operating scenarios to their conventional counterparts. To support the comparison, we developed methodologies and computer tools, which can be used for similar studies in the future. Computer tools in the form of source code are also included as deliverables.

On bus automation, this research developed a new operating concept called a shuttle-centered Automated Bus System and a detailed operating scenario. It also compared the performance of this new concept and scenario with the corresponding light-tail and conventional bus systems. Our numerical results show that the new operating concept has the potential of offering drastic improvement in operational efficiency than its light-rail and conventional counterparts on the mainline operations alone, not to mention on integrated mainline and local operations. Further evaluations and comparisons of these system concepts, with or without integration with the local operations, are worthy future research topics. System simulation may be required. The design of a fail-safe automated driverless closely spaced bus following is a worthy subject for future research. Although routing and scheduling for either bus systems can be optimized, such important tasks require consideration of many factors that are not easy to quantify. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider’s perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

On inter-city trucking automation, this research developed a new operating concept called a Truck-Automated Highway System with shuttle-centered convoying and a detailed operating scenario. It also compared this new concept and scenario with the corresponding general-use-lane and truck-lane alternatives. Our numerical results show that, under the assumptions made and given the corridor selected, the Truck-AHS alternative will not provide any travel time advantage to the overall system or even to the overall trucking industry. The only advantages of truck-AHS are labor and fuel savings, and the labor saving is drastic. These savings must be weighed against the infrastructure costs. Moreover, the safety and technical feasibility of the truck-AHS must be carefully studied. Our comparison suggests that general-use lane is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings.

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

H.-S. Jacob Tsao, Lan Zhang, Lin Lin and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA
San Jose State University

EXECUTIVE SUMMARY

Traffic congestion will continue to worsen and likely worsen at a faster rate than ever. People throughput and freight throughput have become critical issues for California and the rest of the nation. PATH has funded with approximately $125K a research project entitled "Evaluation of Bus and Truck Automation Scenarios" jointly proposed by Jan Botha (Principal Investigator) of Department of Civil and Environmental Engineering and Jacob Tsao (Co-PI) of Department of Industrial and Systems Engineering at San Jose State University. This report summarizes the major findings of the research conducted by Professor Tsao and his assistants with approximately $44K out of the overall funding of $125K for the project; the infrastructure and other aspects of the research are reported by Jan Botha separately.

During the one-year project, we developed detailed operating scenarios for both urban bus automation and inter-city truck automation as well as operating scenarios for conventional alternatives. We also compared the automation operating scenarios to their conventional counterparts. To support the comparison, we developed methodologies and computer tools, which can be used for similar studies in the future. Computer tools in the form of source code are also included as deliverables.

Urban traffic congestion will continue to worsen and likely worsen at a faster rate than ever. People throughput has become a critical issue for all major metropolitan areas. The quality of public transportation services is of primary concern because of its impact on ridership. The operational efficiency of such services is also critical, particularly for metropolitan areas suffering economic slowdown. Various intelligent transportation systems (ITS) technologies have been proposed for improving service quality or efficiency of people throughput. The most technology-intensive among them is bus automation. Several operating concepts describing how automation technologies may help improve people throughput have been proposed. They range from an older concept of bus automated highway system (Bus AHS), i.e., a physically segregated and protected system of highway lane(s) dedicated to the exclusive use by fully-automated buses, to a newer concept of automated bus system (ABUS), i.e., a system of closely-spaced automated bus convoys mimicking the light-rail operations on dedicated right-of-way with at-grade crossings along a busy commute corridor of city streets. The latter was motivated to combine the advantages of light-rail operations with the flexibility of the bus operations.

Based on the concept of ABUS proposed in the Phase I of this research in the previous PATH RFP cycle, this paper proposes a new operating concept that further capitalizes on the flexibility of bus operations. Moreover, the new concept is more complete in that it addresses routing, convoy sizing and scheduling. A basic problem with light-rail operations is the issue of low demand-dependence resulting from the difficulty in dealing with spatial and temporal demand variability. A basic problem with any bus system, automated or conventional, is the transfer issue associated with the need for connection of bus routes in order to go from any arbitrary

point to another. The new concept alleviates the low-demand-dependence issue of light-rail operations and the transfer issue of a bus system. We first address the low-demand-dependence issue of light-rail operation and then the transfer issue.

Due to the track requirement and the bulky and inflexible mechanical connection between light-rail cars, typical light-rail operations involve fixed-length train operations from one end to the other and back. The capacity provided by such operations is independent of the spatial variability of demand along the mainline; it often is insensitive or independent of the temporal variability of demand. This low-demand-dependence issue leads to operational inefficiency.

It is well known that the relative flexibility offered by the use of buses versus the use of light-rail trains includes the use of the same "dual-mode" buses for mainline operations as well as for passenger collection and distribution off the mainline. However, the degree of net benefit hinges upon travel demand and economy of scale. In theory, one or more routes can be offered to meet the demand between any two points off the mainline (and all points along the route(s)). The population density, the activity patterns and the resulting travel demand between such pairs may be low, and, therefore, some hub-and-spoke structure may be required to lower the cost of providing connection from one point to another. However, such operations require transfers. Such transfers tend to increase travel time and reduce travel time reliability significantly, and tend to expose the passengers to the elements and other possible unpleasant situations. This transfer issue may reduce the service quality significantly.

Although the low-demand-dependence issue and the transfer issue are related, we deal with them separately. Also, we limit the scope of this research to the mainline. The low-demand-dependence issue can be addressed with such a focus. The transfer issue for a bus system including off-mainline operations involves a larger context extending beyond the mainline. However, the way the new operating concept works on the mainline can easily reveal how it works in a bus system involving off-mainline operations.

A key advantage of the new bus-automation operating concept, especially when compared to the conventional light-rail operations or ABUS operations, is that the mainline service can be supplied according to the demand with more ease. For example, if a small portion of the mainline enjoys a high demand, more buses can be electronically attached to a bus convoy to satisfy the demand, but the long convoy travels only the portion of high demand and not the entire corridor. This advantage helps alleviate the low-demand-dependence issue. In this concept a bus may travel only part of the mainline, and we refer to the corresponding route a partial route.

Note that this partial route may be a service connecting two stations on the mainline or may be a portion of a route connecting two off-mainline locations. In this paper, we will use the term partial route to refer to the former unless otherwise specified. The way this concept works for the latter can be easily inferred.

A drawback of having partial routes when compared to conventional light-rail (mainline) operations is that a bus may not travel the whole corridor and hence some passengers of a bus may need to transfer to another bus somewhere and somehow so as to reach the desired destination on the corridor. This difficulty is overcome by the new operating concept as follows. With one shuttle bus traveling from one end of the corridor to the other and with every non-shuttle bus *having to* join a shuttle bus (to form a bus convoy or to join such a bus convoy already formed) *as a condition for using* the dedicated right-of-way, passengers destined for any station long the corridor can reach the desired station through intra-convoy transfer(s). Such an intra-convoy transfer requires a change of bus at a station within the same bus convoy and hence requires no waiting at all. Note that in this operating scenario, non-shuttle buses travel only part of the corridor. We refer to this operating concept as "Automated Bus System (ABUS) with shuttle-centered convoying and intra-convoy transfer." Since the feature of intra-convoy transfer of this operating concept is enabled by the feature of shuttle-centered convoying, we refer to this new operating concept simply as "shuttle-centered" ABUS for ease of discussion.

This paper also compares the performance of a shuttle-centered ABUS system with the corresponding light-tail and conventional bus systems. The only difference between a shuttle-centered ABUS and a shuttle-centered (conventional) busway is that the buses in the latter system are driven manually by human drivers and they form bus convoys or clusters with longer inter-bus distances considered safe for manual driving. The focus is on the more tangible aspects of the comparison. In particular, we compare only the mainline operations and do not speculate on the possible amount of ridership increase that could result from the more flexible service. Also, we consider readily quantifiable performance measures like labor requirement, equipment requirement, fuel consumption, and passenger travel time, but do not address the more illusive performance measures like environmental impact.

On the one hand, the proposed concept mimics the train-oriented operations of an urban light-rail system, with *one driver per bus convoy*. On the other hand, it operates more efficiently on the corridor in the sense that flexible use of buses provides *service capacity only where and when it is needed*. The proposed system is more efficient than the corresponding conventional bus system in that, among other things, it requires only one driver per convoy instead of one driver per bus. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider's perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

In this research, we focus on the evaluation and comparison for the mainline operations. Such operations are relative easy to control, and we adopt a deterministic approach. We also focus on new and significant sources of benefits and cost reductions in comparing the alternatives. To make the comparisons more realistic, we focus on a commuter corridor with light-rail service from one end to the other and use the passenger counts riding the Santa Clara County Light Rail operated by Valley Transportation Authority.

Based on these numerical results, it is clear that the shuttle-center ABUS system can be drastically more efficient than the light-rail system in terms of equipment requirement. For example, to satisfy the current demand, only 792 vehicle-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 2040 vehicle-minutes required by the corresponding light-rail operations. This translates into over 60% reduction of equipment requirement.

It is also clear that the shuttle-centered ABUS system is drastically more efficient than the shuttle-centered (conventional) busway. For example, to satisfy the current demand, only 510 driver-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 792 driver-minutes required by the corresponding conventional operations. This translates into over 36% reduction of labor requirement.

While the shuttle-center ABUS offers drastic improvement over the light-rail and shuttle-centered busway operations, its fuel advantage over the shuttle-centered busway ranges from between 2.5% to 5% and hence is not significant.

It is clear that the new operating concept has the potential of offering drastic improvement in operational efficiency than its light-rail and conventional counterparts on the mainline operations alone, not to mention on integrated mainline and local operations. Further evaluations and comparisons of these system concepts, with or without integration with the local operations, are worthy future research topics. System simulation may be required. The design of a fail-safe automated driverless closely spaced bus following is a worthy subject for future research. Although routing and scheduling for either bus systems can be optimized, such important tasks require consideration of many factors that are not easy to quantify. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider's perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

A well known advantage of bus operations when compared to light-rail operations is that the former can integrate local pick-up and local distribution with the mainline operations easily. Quantitative estimation of cost and benefit requires extensive demand modeling, which requires modeling of commuter behavior, equipment and operating costs and a number of other location-specific factors, and hence is beyond the scope of this project. The results of the comparison between ABUS and the conventional alternatives are reported in the first two parts of this report, with the performance measures of equipment requirement, labor requirement, fuel consumption addressed in one part and the disturbance to surrounding traffic addressed in the other. This separation results from the two distinctive approaches and natures of work: a deterministic approach for the first part and the stochastic computer simulation for the second.

On inter-city trucking automation, we developed a new operating concept and a detailed operating scenario. Based on customer needs, stakeholder concerns and available or promising truck-automation technologies, we developed for the Phase I of this research design options for several key aspects of truck-AHS operations, compared the merits of these options, and developed system operating concepts and deployment sequences to satisfy the customer needs. Based on an initial qualitative analysis, the phase-I research developed two operating concepts. Those concepts resemble commercial rail operations in the train-like operations. But they differ from the rail operations considerably in that they are implemented on a long stretch of dedicated and physically separated freeway, possibly occupying the median space of the current interstate freeway system, along a busy freight corridor and that (self-propelled) dual-mode trucks are electronically coupled and organized into convoys while traveling on the freeway, instead of mechanically coupled, and hence can move onto or off from the freeway with ease. We refer to this scenario as Truck Automated Highway System (Truck-AHS). This research developed a new operating concept called Truck-AHS with shuttle-centered convoying; the new concept was motivated to respond to user needs and to increase the deployability while taking advantage of promising technologies.

In this new operating concept, the system is closed in the sense that tractors traveling on the truck-AHS are provided by a small number of operators called "AHS haulers." Also, trucks form a closely-spaced convoy while traveling automatically on the AHS; only the lead truck of the convoy has a human driver, who supervises the operations of the whole convoy. Moreover, convoy merging at an on-ramp and convoy splitting at an off-ramp are both automated. This research further developed this new operating concept to include sufficient operational details for quantitative evaluation and comparison. For example, the Truck-AHS operator runs a shuttle truck from one end of the corridor to the other and back; such a shuttle truck serves as the lead truck of a truck convoy and only such a truck can be the lead truck of a truck convoy. Any automated truck along the corridor that wishes to use the Truck-AHS must join a truck convoy already traveling on the Truck-AHS. The headway of the shuttles is constant throughout the day; the constant headway is determined in such a way that any truck arriving at an Truck-AHS access point can join the next passing truck convoy (after changing modes and waiting for the next passing convoy) and the convoy size does not exceed a prescribed number of trucks, e.g., the limit of 25 trucks in our numerical study. The trucks of a convoy are closely spaced and hence enjoy fuel savings due to reduced air resistance.

We compared the following three specific alternatives in terms of truck travel time, non-truck travel time, trucking labor and fuel consumption:

- General-Use Lane (adding one conventional general-use lane per direction to the conventional freeway)
- Truck-AHS (constructing a physically separated one-lane truck-AHS within or along the right-of-way of a conventional freeway)
- Truck Lane (constructing a physically separated lane dedicated to truck travel within or along the right-of-way of a conventional freeway).

Based on our numerical results, it is clear that, under the assumptions made and given the corridor selected, the Truck-AHS alternative will not provide any travel time advantage to the overall system or even to the overall trucking industry. It is able to provide travel time advantage to only the long-haul trucking industry but at the expense of the short-haul trucking industry and the non-truck driving public. (Trucks using the truck-AHS travel at a consistently higher speed of 75 miles per hour.)

The only advantages of truck-AHS are labor and fuel savings, and the labor saving is drastic. These savings must be weighed against the infrastructure costs. Moreover, the safety and technical feasibility of the truck-AHS must be carefully studied. The Truck Lane alternative does not look promising either; it does not enjoy any advantage over the General-Use Lane alternative, and we have not even begun to address the cost of infrastructure.

Finally, for the General-Use Lane alternative, we compared the addition of only one conventional general-use lane to the other two alternatives. After the "overhead" infrastructure requirements, e.g., the break-down lane or shoulder and the width required for physical barriers, etc., for the other two alternatives are taken into consideration, it is likely that the overall right-of-way required by either of the other two alternatives can accommodate two conventional general-use lanes. As a result, the travel time advantages of the General-Use Lane alternative just reported will likely be clear understatements. In addition, addition of two general-use lanes, without the need for physical separation as required by the other two alternatives, will likely cost drastically less.

Our comparison suggests that general-use lane is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings.

Although these results can provide valuable insights into the advantages and disadvantages of the three alternatives in general, the reader is reminded that what we have evaluated and compared are three specific operational systems and that the comparison is made against a specific reference corridor. In addition, due to the complexity of the problem and the absence of daily origin-destination data for truck trips and time-dependent demand data for freeway sections, several estimation methods have been employed. Caution is needed when generalizing these specific results to the three corresponding general alternatives.

Based on our results and given these limitations, we believe that future studies on truck-AHS should be focused on the feasibility of driverless, automated, closely-spaced truck-following as potential source of labor and fuel savings and on the concomitant infrastructure costs if the purpose of constructing a truck-AHS is to facilitate inter-city trucking. Other truck-AHS operating concepts may benefit other more special purposes.

This report is organized in four parts as follows. Part I describes the project and provides the common background for all the remaining parts. Part II defines the bus automation operating scenario in detail and compares it to three conventional alternatives with respect to three performance measures: equipment requirement, labor requirement and fuel consumption. Part III compares the bus automation scenario to three conventional alternatives with respect to the fourth performance measure: disturbance to surrounding traffic. Part IV defines Truck-AHS, i.e., truck automation operating scenario in detail and compares it to three conventional alternatives with respect to four performance measures: labor requirement, truck travel time, non-truck travel time and fuel savings.

PART I: BACKGROUND AND PROJECT DESCRIPTION

1. BACKGROUND

Traffic congestion will continue to worsen and likely worsen at a faster rate than ever. With the completion of the construction of the National Highway System and the general lack of available right-of-way for adding lanes on existing freeways in the largest metropolitan areas around the nation, the issue of traffic congestion has received more and more attention. Increasing people throughput has become a necessary component of any credible solution to the current and future transportation problems.

California would be the fifth largest economy in the world if it were a nation. Goods movement is a critical component of California's prosperity. Recognizing the importance of goods movement in the state, Caltrans developed the Statewide Goods Movement Strategy as one of the two focal areas of the 1998 California Transportation Plan (CTP) Update (Caltrans, 1998a and 1998b). Increasing the efficiency of goods movement in the state has also become a necessary component of any credible solution to the current and future transportation problems.

Automated driving on freeways has been treated primarily as a means to increase automobile throughput on the nation's highways. The high potential for increasing automobile throughput is accompanied by a high level of risk resulting from the complexity of the technical, institutional and political issues involved in the design of a deployable system and in its staged deployment. The investigator believes that transit- and/or truck-oriented automated highway systems (AHS) could be a more promising concept, not only as an "end-state" by itself but also as an intermediate step toward the implementation of an AHS accommodating also automobiles. The investigator proposed a transit service for AHS debut in a paper published in the IVHS Journal in 1995 (Tsao, 1995d). AHS is the most technology-intensive and forward-looking component of the many intelligent transportation systems (ITS) user services, and it cannot and will not be implemented as a single "bundle" of service features in one giant leap. As a firm believer of incremental deployment of vehicle-automation technologies, the investigator argued for a balanced approach between "market pull" and "technology push," e.g., (Tsao, 1995b), (Tsao, 1998a), (Tsao, 1998b), (Tsao, 2001), (Tsao and Botha, 2001), and (Tsao and Botha, 2003).

To investigate the potential of vehicle automation for significantly improving people throughput and freight throughput of this nation's transportation systems, PATH funded, during the previous yearly PATH funding cycle, a research project entitled "Definition and Evaluation of Bus And Truck Automation Operations Concept" jointly proposed by Jan Botha of Department of Civil and Environmental Engineering and Jacob Tsao of Department of Industrial and Systems Engineering at San Jose State University. The findings of that research have been documented in (Tsao and Botha, 2003). Based on the research of that project, they proposed a second phase of the research as a project entitled "Evaluation of Bus and Truck Automation Scenarios." PATH also funded the second phase, and this report summarizes the major findings of the research conducted by Professor Tsao.

2. PROJECT DESCRIPTION

In this section, we briefly describe

- the scope of research
- related research
- research approach.

2.1 Scope Of Research

The scope of research is as follows:

For Both Bus AHS and Truck AHS:

- A corridor and not a network
- Drastic throughput gain not expected, but treated as a possible goal for the future
- Essential aspects of operations concepts
- Evaluation:
  - Major cost and benefit items
  - Technology: functional specification for the required technology, without any study of technology feasibility
  - Safety: an "intuitive" check of system safety (e.g., fail-safe capability), with safety evaluation set aside for the future
  - Demand as a parameter, with no demand modeling

For Bus AHS:

<u>Relative Cost and benefit of the following key alternatives:</u>

- conventional light-rail system (involving downtown segments)
- an Automated BUs System (ABUS), on current light-rail right of way or on planned new light-rail lines
- busway without automation

For Truck AHS:

<u>Cost and benefit of the following key alternatives</u>

- Adding a conventional lane (without dedication of any lanes to truck use)
- Adding a truck lane
- Adding an exclusive AHS truck lane

2.2 Related Research

Research and implementation efforts on BRT began at least a quarter century ago under the umbrella of dual-mode (bus) transportation. See, for example, (DeMarco, 1974). More recent efforts include the implementation of a BRT system in Adelaide, Australia, (South Australia DOT, 1988) and the study of a guided bus system in Eugene/Springfield area of Oregon (Carey et al., 1998). In the past several years, the subject of BRT took on a broader interpretation as any system that provides some key features of a urban commuter-rail system but with buses, particularly those systems that use advanced technologies to reduce or eliminate impediment to bus movement. The Federal Transit Administration has been providing technical and financial support for the nation's transit agencies to develop BRT systems. For example, (FTA, 1998) summarizes key issues in BRT. In 1998, the FTA issued a Call for Applications for participation in a nationwide BRT implementation effort and has since completed the applicant selection process. The Valley Transit Authority (VTA) of Santa Clara County was one of the ten applicants selected by the FTA. Caltrans facilitated VTA's effort and sponsored a research project through California PATH entitled "Implementation of ITS Technologies for Bus Rapid Transit." Currently, Caltrans is funding a before-and-after study for the implementation of transit signal priority (TSP) along El Camino Real to facilitate the movement of Line 22 buses. This research complements these related research efforts in that it focused on the concept of bus automation. The concept of automated "virtual train" of buses was demonstrated in August 2003, this research puts the concept and the enabling technologies to use in the real world and developed promising operating scenarios for urban bus automation.

Truck automation has also long been considered by many as a promising intermediate step toward an AHS that supports all major vehicle types. It was studied as part of the Precursor System Studies prior to the formation of the National AHS Consortium (NAHSC). It has been an on-going research subject in Europe. The CHAUFFEUR Project has produced promising technologies and cost-benefit findings for truck automation that can be used in developing complete operational concepts and their evaluation. Recent published results include (Baum and Schulz,1997), (Borodani et al., 1997), (Riva and Ulken, 1997), and (Schulze, 1997).

This research project also complements many recent PATH research efforts on truck AHS. Current and recent research sponsored by Caltrans include projects investigating advanced vehicle control of heavy vehicles and the safety and robustness of such control for heavy vehicles, e.g., (Tai, 2001), (Tai, et al., 2001) and (Yip, 2003) and projects investigating into the systems aspects of truck automation, e.g., (Tsao and Botha, 2001; Tsao and Botha, 2002a, Tsao and Botha 2002b; Tsao and Botha 2003). Complementary PATH research projects include FHWA-sponsored projects, e.g., the Automated Highway Systems - Precursor Systems Analyses (AHS-PSA) projects, and many NAHSC-sponsored projects. The vast majority of research papers published by the investigator resulted from Caltrans-sponsored research, either on definition of AHS operating concepts or their evaluation. (The Reference and Bibliography section of this proposal lists only part of the publications.)

2.3 Research Approach

Freeway congestion has been growing steadily, and this trend is projected to continue. Conventional transportation systems have failed to arrest this trend. The concept of automated highway systems (AHS) has received much attention because of its potential of drastically increasing automobile throughput without requiring a significant amount of infrastructure modification. The vast majority of the research attention has been focused on a fully automated high-throughput automobile-AHS, where a system is primarily considered as a vehicle-traffic control system.

AHS research and development is conducted for the ultimate deployment in the real world, and deployment issues are likely to impose constraints on AHS design (Tsao, 1995d). As a result, "system" in this context must include the whole transportation system and the society at large, and deployment issues must be investigated and fully considered at the outset of the AHS R&D process (Tsao, 2001). Some critical issues have not yet been fully addressed, e.g., how to ensure a sufficiently large population of equipped vehicles before opening the fully-automated AHS so as to avoid the so-called "empty-lane syndrome" or "the chicken-and-egg problem", how to deal with failure events, human factor issues, liability issues, etc. As a result, operating concepts that are sustainable have not been developed. With this recognition, Hall and Tsao (Hall and Tsao, 1997) identified many AHS deployment issues, and Tsao (Tsao, 2001) developed a framework for anticipating, recognizing and organizing ITS deployment issues, particularly such issues regarding forward-looking concepts like AHS. Tsao (Tsao, 1995d) discussed critical issues associated with initial deployment of automation technologies and proposed a transit service for AHS debut. He suggested that a bus-AHS could be a goal by itself for high people-throughput or could be an intermediate step toward realizing a fully automated high- throughput automobile-AHS even when the latter is the only goal. AlKadri et al. (AlKadri et al., 1998) and Tsao (Tsao, 1998b) also proposed partial-automation concepts designed to help resolve the "chicken-and-egg" issue associated with AHS deployment. Shladover (Shladover, 2000), extending his earlier work (Shladover, 1999), stated, "The most serious challenge to the credibility of highway automation as a potential solution to transportation problems has been the lack of a convincing deployment strategy." He proposed a set of principles that can be used to guide the design of AHS deployment strategies. He also proposed a set of potential steps beyond adaptive cruise control (ACC) toward an AHS that is protected with barriers and fences. The protection is motivated by the consideration that the driver can no longer be depended on to identify hazards or failures because his or her attentiveness cannot be assured. He also provided example AHS deployment "road maps" for transit buses, heavy trucks and automobiles.

Despite these and other efforts aimed at facilitating AHS deployment, constructing a full-scale bus-AHS network, covering an entire metropolitan area with dedicated right-of-way and new infrastructure, requires a huge investment and strong public will. Something of a smaller scale could acquaint the public with the concept of automation and may help build support for an AHS. A smaller-scale system similar to a light-rail system along a commute corridor may present a feasible opportunity for the deployment of an automated busway (ABUS). An ABUS is any bus system that supports hands-off or feet-off driving.

A well known problem about any rail system is that its success hinges upon a convenient feeder system. The development of most of this nation's metropolitan areas has centered on the use of automobiles as the primary or even the only means of people transportation. The resulting low population density prevents efficient deployment of transit systems. Moreover, when a rail system is implemented, demand for such a system is often inhibited because of the lack of parking at the stations or the nuisance and delay associated with transfer from and to a feeder bus.

An ABUS that has the capability of fulfilling both the speedy line-haul function and local collection/distribution may be a significantly better alternative. In addition, such a corridor ABUS may be a smaller-scale implementation of vehicle-automation technology that can help build the necessary public support for bus automation in particular and for AHS in general. Such a concept may use right-of-way similar to that of a light-rail system for the line-haul proportion. A major functional difference between the ABUS and the light rail system will be that the same buses, engaged in the collection-distribution function, will be used in the line-haul function of the ABUS also, thereby eliminating mode changes.

This advantage is well known but is not easy to quantify because the benefit depends on the usage (change of ridership or change of mode choice) and the usage cannot be accurately predicted without a large-scale research effort encompassing estimation of trip origins and destinations and model development/validation for commuter choice of transportation modes. We focus on the "mainline" when analyzing and comparing the costs and benefits of different alternatives.

Recently, some operating concepts for urban bus automation have been developed, and their advantages discussed in qualitative terms. The major goal of this research is to assess in quantitative terms the cost and benefit of urban bus automation and to compare it to the conventional alternatives also in quantitative terms. For quantitative comparisons, detailed operating scenarios to the level of bus routing and scheduling as a function of travel demand must be developed.

During this one-year project, we developed detailed operating scenarios for urban bus automation as well as operating scenarios for conventional alternatives. We also compared the automation operating scenarios to their conventional counterparts. To support the comparison, we developed methodologies and computer tools, which can be used for similar studies in the future. Computer tools in the form of source code are also included as deliverables.

On urban bus automation, we developed an operating scenario that resembles light-rail operations implemented on stretches of dedicated right-of-way segmented by at-grade crossings along a busy commute corridor but differs considerably from light-rail operations in that dual-mode buses are electronically coupled while traveling on the right-of-way, instead of mechanically coupled, and hence can move onto or off from the right-of-way with ease. We refer to this scenario as Automated BUS System (ABUS) and to the stretches of dedicated right-of-way segmented by at-grade crossings along a busy commute corridor as the "mainline." Other similarities between conventional light-rail operations and ABUS operations include the following. The automated buses move as closely-spaced bus convoys on the mainline at low or medium speeds; the operation of such a convoy is supervised by a human driver in the lead bus for safety reasons. We compared this operating scenario with three conventional alternatives that require the same mainline and are equipped with signal priority or pre-emption at the at-grade crossings: light-rail , busway and busway implemented with Intelligent Transportation Systems (ITS) technology for clustering buses (to avoid excessive disturbance to surrounding traffic).

This research has focused on new and significant sources of benefits and cost reductions in comparing the alternatives. To make the comparisons more realistic, we focus on a commuter corridor with light-rail service from one end to the other and use the passenger counts riding the Santa Clara County Light Rail operated by Valley Transportation Authority.

A key advantage of this bus automation scenario is that the service can be supplied according to the demand with ease. For example, if a small portion of the mainline enjoys a high demand, more buses can be electronically attached to a bus convoy to satisfy the demand, but the long convoy travels only the portion of high demand. Due to the track requirement and the bulky and inflexible mechanical connection between light-rail cars, typical light-rail operations involve fixed-length train operations from one end to the other and back. However, a major drawback of this automated bus operation when compared to conventional light-rail operations is that a bus may not travel the whole corridor and hence some passengers of a bus may need to transfer to another bus somewhere and somehow so as to reach the desired destination on the corridor. This difficulty is overcome by the operating scenario. With one shuttle bus traveling from end to end and every other bus joining a shuttle bus to form a bus convoy and to join such a bus convoy, passengers destined for any station long the corridor can reach the desired station through intra-convoy transfer(s) and such intra-convoy transfers require a change of bus within the same bus convoy and hence require no waiting at all. Note that in this operating scenario, buses other than the end-to-end shuttle buses travel only part of the corridor.

We refer to this operating scenario as "shuttle-centered" automated bus convoys with intra-convoy passenger transfer or simply as "shuttle-centered" ABUS with intra-convoy transfer. We refer to the portion of the mainline served by a non-shuttle bus a "partial route." Note that a bus route may begin at a location off the mainline or end at a location off the mainline so as to perform local collection and distribution in addition to the mainline service and hence that the partial route of the bus along the mainline is part of the overall route from or to the off-mainline end points. More importantly, when such routes are implemented with the "shuttle-centered" ABUS with intra-convoy transfer, transfer from one bus route to another involves no waiting due to connection and minimum exposure to the elements.

Measures of performance for comparing ABUS with the conventional alternatives include the equipment requirement, labor requirement, passenger travel time and disturbance to surrounding traffic. Our results demonstrate that ABUS is significantly better than all the other conventional alternatives in at least one of the four performance measures. A well known advantage of bus operations when compared to light-rail operations is that the former can integrate local pick-up and local distribution with the mainline operations easily. Quantitative estimation of cost and benefit requires extensive demand modeling, which requires modeling of commuter behavior, equipment and operating costs and a number of other location-specific factors, and hence is beyond the scope of this project. The results of the comparison between ABUS and the conventional alternatives are reported in the first two parts of this report, with the performance measures of equipment requirement, labor requirement, fuel consumption addressed in one part and the disturbance to surrounding traffic addressed in the other. The separation results from the two distinctive approaches and natures of work: a deterministic approach for the first part and the stochastic computer simulation for the second.

Like bus automation, truck automation has also been viewed by many as a viable goal by itself or as an intermediate step toward a fully automated high-automobile-throughput AHS. This report also discusses a truck-AHS that operates on a barrier-separated and dedicated lane on a freeway along an inter-city freight corridor where sufficient demand and right-of-way exist.

On inter-city trucking automation, we developed a new operating concept and a detailed operating scenario. Based on customer needs, stakeholder concerns and available or promising truck-automation technologies, we developed for the Phase I of this research design options for several key aspects of truck-AHS operations, compared the merits of these options, and developed system operating concepts and deployment sequences to satisfy the customer needs. Based on an initial qualitative analysis, the phase-I research developed two operating concepts. Those concepts resemble commercial rail operations in the train-like operations. But they differ from the rail operations considerably in that they are implemented on a long stretch of dedicated and physically separated freeway, possibly occupying the median space of the current interstate freeway system, along a busy freight corridor and that (self-propelled) dual-mode trucks are electronically coupled and organized into convoys while traveling on the freeway, instead of mechanically coupled, and hence can move onto or off from the freeway with ease. We refer to this scenario as Truck Automated Highway System (Truck-AHS). This research developed a new operating concept called Truck-AHS with shuttle-centered convoying; the new concept was motivated to respond to user needs and to increase the deployability while taking advantage of promising technologies.

We compared this operating scenario with two conventional alternatives: adding a general-use lane and adding a conventional truck lane. Measures of performance for comparing Truck-AHS with the conventional alternatives include the labor requirement, truck travel time, non-truck travel time and fuel savings. To make the comparison more realistic, we use recent traffic demand along Interstate 5 in California. Comparisons are also made for inflated traffic demand. Our results demonstrate that Truck-AHS is significantly better than all the other conventional alternatives in at least one of the four performance measures. The results of the comparisons between Truck-AHS and the conventional alternatives are reported in the third part of this report. The unit costs of these four measures as well as the unit costs of the different infrastructure requirements are reported separately by Professor Jan Botha. The performance measures obtained in this research first weighted by those unit costs and then normalized by time value of money reveal overall relative costs and benefits of the different alternatives.

2.4 Organization of This Report

This report is organized in four parts as follows. Part I so far describes the project and provides the common background for all the remaining parts. Part II defines the bus automation operating scenario in detail and compares it to three conventional alternatives with respect to three performance measures: equipment requirement, labor requirement and fuel consumption. Part III compares the bus automation scenario to three conventional alternatives with respect to the fourth performance measure: disturbance to surrounding traffic. Part IV defines Truck-AHS, i.e., truck automation operating scenario in detail and compares it to three conventional alternatives with respect to four performance measures: labor requirement, truck travel time, non-truck travel time and fuel savings.

REFERENCES

Al-Ayat, R., and Hall, R.W., “A conceptual approach for developing and analyzing alternate evolutionary deployment strategies for Intelligent Vehicle/Highway Systems,” PATH Working Paper 94-5, 1994.

AlKadri, M., Benouar, H., and Tsao, H.-S.J., "Intermediate Automation Concepts for Evolution toward Automated Highway Systems," Transportation Research Record No. 1651, 1998.

Bagby, J.W., and Gittings, G.L., “Litigation Risk Management for Intelligent Transportation Systems,” ITS Quarterly, Spring-Summer, 1999.

Baum, H. and Schulz, W.H., “Economic Evaluation of CHAUFFEUR,” Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Black, Alan. *Urban Mass Transit Planning*. McGraw-Hill, 1995.

Borodani, P., Carrea, P., and Gortan, L., "Short headway Control of Trucks on Motorways: the CHAUFFEUR Project," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

California Department of Motor Vehicles, California Commercial Driver Handbook, 2001.

California Legislative Counsel, California Vehicle Code, Section 35000, Division 15, Sacramento, California. (http://www.leginfo.ca.gov/.html/veh_table_of_contents.html)

Calspan Corporation, Precursor Systems Analyses of Automated Highway Systems Commercial and Transit AHS Analysis. Vol. VII, FHWA-RD-95-134, Federal Highway Administration, U.S. Department of Transportation, 1995.

Caltrans (1998a), 1998 California Transportation Plan: Statewide Goods Movement Strategy, Office of the Director, California Department of Transportation, Sacramento, California, October,1998.

Caltrans (1998b), 1998 California Transportation Plan: Transportation System Performance Measures (Final Report), Office of the Director, California Department of Transportation, Sacramento, California, August,1998.

California PATH, "White Paper: Development of BRT Evaluation Tools and Techniques," submitted to the Federal Transit Administration in response to the FTA's interest, 1999.

Cambridge Systematics, Inc. *Characteristics of Urban Transportation Systems. Final Report.* Federal Transit Administration, Washington D.C., September 1992.

Carey, G. N., Viggiano, S. M., and Gardner, L., "The development of a guided bus system : the Eugene/Springfield experience," Institute of Transportation Engineers. Meeting (68th : 1998 : Toronto, Ont.). Annual meeting papers, Institute of Transportation Engineers, Washington DC.

Chen, C.-Y. and Litkouhi, B., "Technological Challenges in the Development of AHS," ITS Journal, Vol. 4, No. 1-2, pp. 81-100, 1998.

DeMarco, V. R., "Dual-Mode System Development Program of the Urban Mass Transportation Administration," Proceedings of International Conference on Dual Mode Transportation, Washington, D.C., 1974; Transportation Research Board Special Report No. 170, **:** Transportation Research Board, National Research Council, Washington, D.C..

Federal Highway Administration, Comprehensive Truck Size and Weight Study, U.S. Department of Transportation, 2000.

Federal Transit Administration, "Issues in Bus Rapid Transportation," U.S. Department of Transportation, FTA, Office of Research, Demonstration and Innovation, 1998.

Federal Transit Administration. National Transit Database 1998. http://www.fta.dot.gov/ntl/database.html.

Hall, R.W., "System Configurations: Evolutionary Deployment Considerations," Chapter 4 of Automated Highway Systems, P.A. Ioannou (editor), Plenum Press, New York, pp. 49-71, 1997.

Hall, R.W., Nowroozi, A., and Tsao, H.-S.J., "Entrance Capacity of an Automated Highway System," to *Transportation Science*, Vol. 35, No. 1, pp. 19 – 36, 2001.

Hall, R.W. and Tsao, H.-S. J. "Automated Highway System Deployment: A Preliminary Assessment of Uncertainties", Chapter 16 of Automated Highway Systems, P.A. Ioannou (editor), Plenum Press, New York, pp. 325 - 334, 1997.

Hall, R.W., and H.-S. J. Tsao "Capacity of Automated Highway Systems: Merging Efficiency," Proceedings of American Control Conference, Albuquerque, New Mexico, June, 1997, pp. 2046-2050.

Hall, R.W., Thakker,V., Horan, T.A., Glazer, J., and Hoene, C., Automated Highway System Field Operational Tests for the State of California: Potential Sites, Configurations and Characteristics, California PATH Research Report UCB-ITS-PRR-97-45, 1997.

Hanson, M. and H.-S.J. Tsao, "Leveraging Exogenous Events for the Deployment of Automated Highway Systems," Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida (1996).

International Conference on Truck Weight and Size Limits, Canada, 1989.

KAKU Associates, SR-60 Truck Lane Feasibility Study – Final Report, prepared for Southern California Association of Governments, Feb. 2001.

Kawamura, K., Commercial Vehicle Value of Time and Perceived Benefit of Congestion Pricing, Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of California, Berkeley, 1999.

Lankard, T.L., and Lehrer, J., “Big Trucks, Big Trouble ? – Long Combination Vehicles Could Mean Greater Risks on California Roads”, Westways, Vol. 91, No.6, Auto Club of Southern California, 1999.

Levinson, Herbert S.; Zimmerman, Samuel; Clinger, Jennifer; Gast, James, “Bus rapid transit : synthesis of case studies,” Transportation research record. No. 1841, p. 1-11, 2003.

Levinson, Herbert S., Bus Rapid Transit, Volume 2: Implementation Guidelines, Transit Cooperative Research Program Report #90, Transportation Research Board, U.S.A., 2003.

March, J.W., “DOT’s Comprehensive Truck Size and Weight Study – A Summary,” Public Roads, March/April 2001 Issue, Turner-Fairbank Highway Research Center, 2001.

McKenzie, D.R., North, Mark C., and Smith, D.S., Intermodal Transportation – The Whole Story, Simmons-Boardman Books, Inc., Omaha, Nebraska, August, 1989.

McCormick, E.R., A 12-14-1999 Letter From ATA President and CEO McCormick to President and CEO of AAR (to ask for a summit on truck productivity), Inside ATA, American Trucking Associations, Dec. 15, 1999.
(http://www.trucking.org/insideata/outreach/121599_ltr_railroad.html.)

Michael, J.B., Godbole, D.N., Lygeros, J., and Sengupta, R., “Capacity Analysis of Traffic Flow Over a Single-Lane Automated Highway System,” ITS Journal, Vol. 4, No. 1-2, pp. 49-80,1998.

Miller, M.A. and Tsao, H.-S.J., “Testing a Decision-Oriented Framework to Understand ITS Deployment Issues: Lessons Learned from the TravInfo ATIS Project,” Proceedings of the 6th World Congress on Intelligent Transport Systems, Toronto, Canada, November 8-12, 1999.

Ran, B., Johnson, S., Leight., S, and Tsao, H.-S.J., "Traffic Analysis for Highway-to-Highway Interchanges on Automated Highway Systems: Congestion in Absence of Dedicated Ramps", Transportation Research Record 1588: Planning and Administration; Highway Operations, Capacity, and Traffic Control, pp. 145-151, 1997.

Riva, P. and Ulken, U., "CHAUFFEUR Tow-Bar: the Vehicle Based System Architecture," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Schulze, M., "CHAUFFEUR, the European Way towards an Automated Highway System," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Sengupta, R., Godbole, D. and H.-S.J. Tsao, "Concept Definition of an Infrastructure-Supported Automated Highway System," Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida,1996.

Shladover, S.E., "Review of State of Development of Advanced Vehicle Control Systems (AVCS)," International Journal of Vehicle Mechanics and Mobility, Vol. 24, pp. 551-595, 1995.

Shladover, S.E., "Progressive Deployment of Vehicle-Highway Automation Systems," Proceedings of the 6th World Congress on Intelligent Transportation Systems, ITS America, Toronto, Ontario, Canada, 1999.

Shladover, S.E., "Progressive Deployment Steps Leading Toward an Automated Highway System," Transportation Research Record 1727, pp. 154 - 161, Transportation Research Board, 2000.

Smith, Daniel S. (Principal Investigator), Double Stack Container Systems: Implications for U.S. Railroads and Ports, Research Report FRA-RRP-90-1, Federal Railroad Administration, June, 1990.

Solomon, D., Accidents on Main Rural Highways Related to Speed, Driver and Vehicle, US Department of Transportation, Federal Highway Administration, Washington, D.C., 1974.

South Australia. Dept. of Transport, Guided bus rapid transit : proceedings, international seminar, Adelaide, South Australia, 1988.

State of California, California Vehicle Code, Section 35000, Division 15. California Legislative Counsel, Sacramento, California, 2001.
(http://www.leginfo.ca.gov/.html/veh_table_of_contents.html)

Stevens, W.B., "Evolution to an Automated Highway System," Chapter 7 of 4 of Automated Highway Systems, P.A. Ioannou (editor), Plenum Press, New York, pp. 109-124, 1997.

Tai, Meihua, Advanced vehicle control of heavy vehicles for automated highway systems, Ph.D. Thesis, Department of Mechanical Engineering, University of California, Berkeley, 2001.

Tai, Meihua; Wang, Jeng-Yu; White, Ryan; Tomizuka, Masayoshi, Robust lateral control of heavy duty vehicles, PATH research report ; UCB-ITS-PRR-2001-35, University of California, Berkeley. Institute of Transportation Studies Partners for Advanced Transit and Highways (Calif.) University of California, Berkeley, Dec. 2001.

Transportation Research Board, An Investigation of Truck Size and weight Limits: Panel Review of Final Report (including comments made by major stakeholders, e.g., American Trucking Associations, American Automobile Association, Environmental Defense Fund), Office of the Secretary, U.S. Department of Transportation, UR 26, June 1982.

Tsao, H.-S. J. (1995a), "Traffic Control for Automated Highway Systems: A Conceptual Framework", Transportation Research, Part C, Vol. 3, No. 4, pp. 227-246, 1995.

Tsao, H.-S. J. (1995b), "Stage Definition for AHS Deployment and an AHS Evolutionary Scenario", Intelligent Vehicle Highway Systems Journal, Vol.2(4), pp. 359-382, 1995.

Tsao, H.-S. J. (1995c), "A Staggered-Diamond Design for Automated/Manual-HOV Highway-to-Highway Interchanges and Constraints on AHS Design for Accommodating Automated Highway Change", Intelligent Vehicle Highway Systems Journal, Vol.2(3), pp. 281-292, 1995.

Tsao, H.-S. J. (1995d), "Constraints on Initial AHS Deployment and the Concept Definition of a Shuttle Service for AHS Debut", IVHS Journal, Vol. 2(2), 159-173, 1995.

Tsao, H.-S. J. (1998a), "From the Guest Editor," Special Issue of the ITS Journal on Automated Highway Systems, Vol. 4, Nos:1-2, 1998.

Tsao, H.-S.J. (1998b), "An Axiomatic Approach to Developing Partial Automation Concepts for Deployment of Automated Highway Systems and Partial Invocation of Vision-Based Lane-Keeping and Adaptive Cruise Control," *Transportation Research Record*, No. 1651, 1998.

Tsao, H.-S. J. (1998c), "The Role of Air Cargo in California's Goods Movement,", UCB-ITS Research Report UCB-ITS-RR-98-7, University of California at Berkeley, Institute of Transportation Studies, Sept. 1998.

Tsao, H.-S. J. (2001), " A Framework for Evaluating Deployment Strategies for Intelligent Transportation Systems", Intelligent Transportation Systems Journal (ITS Journal), Vol. 6, pp. 141-173, 2001.

Tsao, H.-S.J. and Botha, J.L. (2001), “Operating Concepts for Urban Bus Automation and Inter-city Truck Automation,” the Proceedings of the 2001 IEEE ITS Conference, Oakland, California.

Tsao, H.-S.J. and Botha, J.L (2002a) , “An Automated Highway System Dedicated To Inter-City Trucking: Operating Concepts And Deployment,” presented at the 2002 Annual Meeting of the Transportation Research Board on Jan. 14, 2002, Washington, D.C.

Tsao, H.-S.J. and Botha, J.L. (2002b), “An Automated Highway System Dedicated To Inter-City Trucking: Design Options, Operating Concepts And Deployment,”, Intelligent Transportation Systems Journal (The ITS Journal), Vol. 7, No. 2, pp. 169 – 196, 2002.

Tsao, H.-S. J., Botha, J.L. (2003), “Definition and Evaluation of Bus and Truck Automation Operations Concepts: Final Report,” California PATH Research Report UCB-ITS-PRR-2003-19, May 2003.

Tsao, H.-S. J. and Hall, R.W., "A Probabilistic Model for AVCS Longitudinal Collision/Safety Analysis", Intelligent Vehicle Highway Systems Journal, Vol.1, No. 3, pp. 261-274, 1994.

Tsao, H.-S. J., Hall, R.W., and Chatterjee, I., "Analytical Models for Vehicle/Gap Distribution on Automated Highway Systems", Transportation Science, Vol. 31, No. 1., pp. 18-33, 1997.

Tsao, H.-S. J., R.W. Hall and S.E.Shladover, "Design Options for Operating Automated Highway Systems", Proceedings of Vehicle Navigation & Information Systems Conference (Oct. 1993).

Tsao, H.-S.J., Hall, R.W., Shladover, S.E., Plocher, T.A. and Levitan, L.J., "Human Factors Design of Automated Highway Systems: First Generation Scenarios,” US-DOT FHWA Report FHWA-RD-93-123 (1993).

Tsao, H.-S. J.,, T.A. Plocher, W.B. Zhang, and S.E. Shladover, "Human Factors Design for Automated Highway Systems: Second Generation Scenarios", draft US-DOT FHWA Report, under FHWA Contract No. DTFH61-91-C-00100, (Jan. 14, 1994).

Tsao, H.-S. J. and Ran, B., "Driver Intelligence Replacement in a Decision-Oriented Deployment Framework for Driving Automation", Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida (1996).

Tsao, H.-S. Jacob and Rizwan, Asim, The Role of Intelligent Transportation Systems in Intermodal Air Cargo Operations," Research Report UCB-ITS-RR-2000-5, Institute of Transportation Studies, University of California, Berkeley, 2000.

Ulmer, Berthold, "CHAUFFEUR: From Concept to Realization," 3rd International Workshop on Vehicle-Highway Automation, Toronto, Canada, Nov. 11-12, 1999.

U.S. Department of Transportation, U.S. DOT Press Release: U.S. DOT Determined to Move Hours-of-Service Regulation Forward, Will Extend Time for Public Comment (06/09/00), http://www.fmcsa.dot.gov/contactus/press/2000/060900.htm.

Yih, Paul; Satyan, Krishna; Gerdes, J. Christian, Safety performance and robustness of heavy vehicle AVCS, PATH research report ; UCB-ITS-PRR-2003-5, University of California, Berkeley. Institute of Transportation Studies Partners for Advanced Transit and Highways (Calif.), February 2003.

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

PART II - EVALUATION OF URBAN BUS AUTOMATION: EQUIPMENT, LABOR AND FUEL CONSUMPTION

AN AUTOMATED BUS SYSTEM WITH SHUTTLE-CENTERED CONVOYING AND INTRA-CONVOY TRANSFER: OPERATIONS AND EVALUATION

H.-S. Jacob Tsao, Lan Zhang and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA

TABLE OF CONTENTS

AN AUTOMATED BUS SYSTEM WITH SHUTTLE-CENTERED CONVOYING AND INTRA-CONVOY TRANSFER: OPERATIONS AND EVALUATION

H.-S. Jacob Tsao, Lan Zhang and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA

Abstract

Various intelligent transportation systems (ITS) technologies have been proposed for improving service quality or efficiency of people throughput. The most technology-intensive among them is bus automation. The recent concept of automated bus system (ABUS), i.e., a system of closely-spaced automated bus convoys mimicking the light-rail operations on dedicated right-of-way with at-grade crossings along a busy commute corridor of city streets, was motivated to combine the advantages of light-rail operations with the flexibility of the bus operations. This paper proposes a new operating concept that further capitalizes on the flexibility of bus operations and can achieve higher operational efficiency than light-rail for the mainline operations alone, without even considering the potential for the efficiency gain when integrated with local collection and distribution. A basic problem with light-rail operations is the issue of low demand-sensitivity resulting from the difficulty in dealing with spatial and temporal demand variability. A basic problem with any bus system, automated or conventional, is the transfer issue associated with the need for connection of bus routes in order to go from any arbitrary point to another. The new concept alleviates the low-demand-sensitivity issue of light-rail operations and the transfer issue of a bus system. We focus on the mainline operations and do not address off-mainline operations in our quantitative evaluation and comparison. Such a focus allows a deterministic approach. Our results show that the new concept can be drastically more efficient than light-rail operations.

AN AUTOMATED BUS SYSTEM WITH SHUTTLE-CENTERED CONVOYING AND INTRA-CONVOY TRANSFER: OPERATIONS AND EVALUATION

H.-S. Jacob Tsao, Lan Zhang and Deepa Batni
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA

Executive Summary

Urban traffic congestion will continue to worsen and likely worsen at a faster rate than ever. People throughput has become a critical issue for all major metropolitan areas. The quality of public transportation services is of primary concern because of its impact on ridership. The operational efficiency of such services is also critical, particularly for metropolitan areas suffering economic slowdown. Various intelligent transportation systems (ITS) technologies have been proposed for improving service quality or efficiency of people throughput. The most technology-intensive among them is bus automation. Several operating concepts describing how automation technologies may help improve people throughput have been proposed. They range from an older concept of bus automated highway system (Bus AHS), i.e., a physically segregated and protected system of highway lane(s) dedicated to the exclusive use by fully-automated buses, to a newer concept of automated bus system (ABUS), i.e., a system of closely-spaced automated bus convoys mimicking the light-rail operations on dedicated right-of-way with at-grade crossings along a busy commute corridor of city streets. The latter was motivated to combine the advantages of light-rail operations with the flexibility of the bus operations.

Based on the concept of ABUS, this paper proposes a new operating concept that further capitalizes on the flexibility of bus operations. Moreover, the new concept is more complete in that it addresses routing, convoy sizing and scheduling. A basic problem with light-rail operations is the issue of low demand-dependence resulting from the difficulty in dealing with spatial and temporal demand variability. A basic problem with any bus system, automated or conventional, is the transfer issue associated with the need for connection of bus routes in order to go from any arbitrary point to another. The new concept alleviates the low-demand-dependence issue of light-rail operations and the transfer issue of a bus system. We first address the low-demand-dependence issue of light-rail operation and then the transfer issue.

Due to the track requirement and the bulky and inflexible mechanical connection between light-rail cars, typical light-rail operations involve fixed-length train operations from one end to the other and back. The capacity provided by such operations is independent of the spatial variability of demand along the mainline; it often is insensitive or independent of the temporal variability of demand. This low-demand-dependence issue leads to operational inefficiency.

It is well known that the relative flexibility offered by the use of buses versus the use of light-rail trains includes the use of the same "dual-mode" buses for mainline operations as well as for passenger collection and distribution off the mainline. However, the degree of net benefit hinges upon travel demand and economy of scale. In theory, one or more routes can be offered to meet the demand between any two points off the mainline (and all points along the route(s)). The population density, the activity patterns and the resulting travel demand between such pairs may be low, and, therefore, some hub-and-spoke structure may be required to lower the cost of providing connection from one point to another. However, such operations require transfers. Such transfers tend to increase travel time and reduce travel time reliability significantly, and tend to expose the passengers to the elements and other possible unpleasant situations. This transfer issue may reduce the service quality significantly.

Although the low-demand-dependence issue and the transfer issue are related, we deal with them separately. Also, we limit the scope of this research to the mainline. The low-demand-dependence issue can be addressed with such a focus. The transfer issue for a bus system including off-mainline operations involves a larger context extending beyond the mainline. However, the way the new operating concept works on the mainline can easily reveal how it works in a bus system involving off-mainline operations.

A key advantage of the new bus-automation operating concept, especially when compared to the conventional light-rail operations or ABUS operations, is that the mainline service can be supplied according to the demand with more ease. For example, if a small portion of the mainline enjoys a high demand, more buses can be electronically attached to a bus convoy to satisfy the demand, but the long convoy travels only the portion of high demand and not the entire corridor. This advantage helps alleviate the low-demand-dependence issue. In this concept a bus may travel only part of the mainline, and we refer to the corresponding route a partial route.

Note that this partial route may be a service connecting two stations on the mainline or may be a portion of a route connecting two off-mainline locations. In this paper, we will use the term partial route to refer to the former unless otherwise specified. The way this concept works for the latter can be easily inferred.

A drawback of having partial routes when compared to conventional light-rail (mainline) operations is that a bus may not travel the whole corridor and hence some passengers of a bus may need to transfer to another bus somewhere and somehow so as to reach the desired destination on the corridor. This difficulty is overcome by the new operating concept as follows. With one shuttle bus traveling from one end of the corridor to the other and with every non-shuttle bus *having to* join a shuttle bus (to form a bus convoy or to join such a bus convoy already formed) *as a condition for using* the dedicated right-of-way, passengers destined for any station long the corridor can reach the desired station through intra-convoy transfer(s). Such an intra-convoy transfer requires a change of bus at a station within the same bus convoy and hence requires no waiting at all. Note that in this operating scenario, non-shuttle buses travel only part of the corridor. We refer to this operating concept as "Automated Bus System (ABUS) with shuttle-centered convoying and intra-convoy transfer." Since the feature of intra-convoy transfer of this operating concept is enabled by the feature of shuttle-centered convoying, we refer to this new operating concept simply as "shuttle-centered" ABUS for ease of discussion.

This paper also compares the performance of a shuttle-centered ABUS system with the corresponding light-tail and conventional bus systems. The only difference between a shuttle-centered ABUS and a shuttle-centered (conventional) busway is that the buses in the latter system are driven manually by human drivers and they form bus convoys or clusters with longer inter-bus distances considered safe for manual driving. The focus is on the more tangible aspects of the comparison. In particular, we compare only the mainline operations and do not speculate on the possible amount of ridership increase that could result from the more flexible service. Also, we consider readily quantifiable performance measures like labor requirement, equipment requirement, fuel consumption, and passenger travel time, but do not address the more illusive performance measures like environmental impact.

On the one hand, the proposed concept mimics the train-oriented operations of an urban light-rail system, with *one driver per bus convoy*. On the other hand, it operates more efficiently on the corridor in the sense that flexible use of buses provides *service capacity only where and when it is needed*. The proposed system is more efficient than the corresponding conventional bus system in that, among other things, it requires only one driver per convoy instead of one driver per bus. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider's perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

In this research, we focus on the evaluation and comparison for the mainline operations. Such operations are relative easy to control, and we adopt a deterministic approach. Our numerical results indicate that the new operating concept has the potential of offering drastic improvement in operational efficiency than its light-rail and conventional counterparts on the mainline operations alone, not to mention on integrated mainline and local operations.

Based on these numerical results, it is clear that the shuttle-center ABUS system can be drastically more efficient than the light-rail system in terms of equipment requirement. For example, to satisfy the current demand, only 792 vehicle-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 2040 vehicle-minutes required by the corresponding light-rail operations. This translates into over 60% reduction of equipment requirement.

It is also clear that the shuttle-centered ABUS system is drastically more efficient than the shuttle-centered (conventional) busway. For example, to satisfy the current demand, only 510 driver-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 792 driver-minutes required by the corresponding conventional operations. This translates into over 36% reduction of labor requirement.

While the shuttle-center ABUS offers drastic improvement over the light-rail and shuttle-centered busway operations, its fuel advantage over the shuttle-centered busway ranges from between 2.5% to 5% and hence is not significant.

It is clear that the new operating concept has the potential of offering drastic improvement in operational efficiency than its light-rail and conventional counterparts on the mainline operations alone, not to mention on integrated mainline and local operations. Further evaluations and comparisons of these system concepts, with or without integration with the local operations, are worthy future research topics. System simulation may be required. The design of a fail-safe automated driverless closely spaced bus following is a worthy subject for future research. Although routing and scheduling for either bus systems can be optimized, such important tasks require consideration of many factors that are not easy to quantify. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider's perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

AN AUTOMATED BUS SYSTEM WITH SHUTTLE-CENTERED CONVOYING AND INTRA-CONVOY TRANSFER: OPERATIONS AND EVALUATION

1. INTRODUCTION

Urban traffic congestion will continue to worsen and likely worsen at a faster rate than ever. People throughput has become a critical issue for all major metropolitan areas. The quality of public transportation services is of primary concern because of its impact on ridership. The operational efficiency of such services is also critical, particularly for metropolitan areas suffering economic slowdown. Various intelligent transportation systems (ITS) technologies have been proposed for improving service quality or efficiency of people throughput. The most technology-intensive among them is bus automation.

1.1 A Brief Literature Review

The most forward-looking user service of Intelligent Transportation Systems is automated highway systems (AHS). AHS was originally conceived as a way to increase vehicle throughput and relieve traffic congestion on urban freeways significantly without requiring significant acquisition of right-of-way. The focus has been on *full* automation of *automobiles* since the very beginning of the research.

Concomitant with the throughput potential are (a) the complex technology requirements, e.g., replacing the adaptable human in performing usual driving tasks and in responding to unusual events with a reliable yet affordable technology (Tsao et al. 1993; Tsao et al., 1994; Tsao and Ran, 1996), and (b) the challenging deployment issues, e.g., the "chicken-and-egg" issue (Tsao, 1998a; Tsao, 1995a; Tsao, 1995b; Tsao, 1995c; Tsao, 1998b; Tsao, 2001).

AHS research, development and deployment for a *fully automated* system has at least two very difficult hurdles to overcome. First, full automation by nature reduces driver alertness or even disengages the driver, and, therefore, the driver should not be expected to be able to deal with any abnormal events by first taking over driving and then responding to the events. This leads to the necessity for the designer of a fully automated system to anticipate major failure and other abnormal events, the companion necessity for such a system to deal with the abnormal events safely, and also the necessity to disallow driver intervention in dealing with such abnormal events. Operating concepts for several such fully automated systems were developed in the context of human-system interaction (Tsao et al., 1993), and numerous major failure events were identified (Tsao et al., 1994). This critical issue, i.e., a potential "show-stopper", was identified before the formation of the National Automated Highway Systems Consortium (NAHSC), but did not seem to have received any attention prior to the formation of the NAHSC or within the shortened life of NAHSC. Shladover (1995) surveyed the extensive research efforts and results about advanced vehicle control systems (AVCS). Vehicle control is a component of the task of safe control of a fully automated highway system. Based on their extensive experience in system design for automated highway systems, Chen and Litkouhi (1998) discussed the challenges facing the designers of a fully automated highway system. In parallel to the NAHSC activities, a small amount of research dealt with this issue in a systematic way. For example, Tsao and Ran (1996) posed the fundamental question of what constitutes “freeway driving”, including what a driver usually does and what he or she may encounter occasionally or rarely, proposed to identify systematically the technologies required to replace the human driver so that automated driving is at least as safe as human driving on freeways, and also proposed to assess the technical and economical feasibility of such full automation. Note that possible liability issues exist even when the system can perform freeway driving at exactly the same level of safety as human driving. This is because the driver will be disallowed to intervene in any abnormal events and they may put the blame of the result on the infrastructure operator or the vehicle manufacturer.

Second, one way to reduce the technological complexity is to physically separate the fully automated traffic from the conventional traffic. However, this requires major infrastructure modification and construction and even major right-of-way acquisition. First of all, this defeats the purpose and promise of AHS. In addition, to enable continuous automated driving from one freeway to another, an additional set of connector ramps will be required even if the new automated lanes are to be built on current freeway right-of-way (Tsao, 1995b). Moreover, if the system is to be implemented in a revolutionary way as promoted by the NAHSC and its strong supporters, instead of an incremental fashion, the major providers of such a system, e.g., infrastructure provider, vehicle manufacturer, insurance provider, etc., will likely be caught in a deadlock, each of which will be waiting for the others to act boldly first. These issues were also identified before the formation of the NAHSC. Few efforts were devoted to resolving this issue. For more information about this issue or the efforts, the reader is referred to (Tsao, 1995a; Tsao, 1995b; Tsao, 1995c; Tsao, 1998a; Tsao, 1998b; Shladover, 2000; Tsao, 2001).

The fact that the National Automated Highway Systems Consortium (NAHSC) was not able to deal with these technology and deployment issues and hence was discontinued prematurely does not mean that automation technologies cannot benefit surface transportation. Interest in automated bus operations started at least three decades ago (DeMarco , 1974). Tsao (1995c) proposed a concept of automated freeway shuttle bus for AHS debut. Bishop (2000) reported increasing interest in automated bus operations within the Bus Rapid Transit (BRT) Consortium recently promoted by the U.S. Federal Transit Administration. Recent efforts on the technical aspects of bus automation include Aso and Suzuki (2000),

The California PATH Program demonstrated key technologies for automated bus rapid transit (A-BRT). (Shladover, 2003; Tan, 2003). Three transit buses were equipped with sensing, actuation, communication and computation systems. They were also equipped with a driver-vehicle interface (DVI) system allowing the driver to interact with the automation systems, transfer back and forth between manual driving and automation, and initiate automated maneuvers. The demonstrated transit service functions included:

precision docking, automatic lane-keeping, automatic lane-changing, fully automated bus driving, and automated "virtual train", i.e., automated bus convoy.

The previous phase, i.e., Phase I, of this research was motivated to (a) identify opportunities for automation technologies, particularly *partial-automation* technologies, to benefit *heavy-vehicle operations*, (b) develop operating concepts in sufficient detail for specifying *vehicle and infrastructure functional requirements*, and (c) develop *deployment sequences* for these operating concepts.

The purposes of this research are to (a) further develop sufficient *operational details* for automation concepts proposed in Phase I in order to perform cost-benefit evaluations and comparisons with conventional alternatives and (b) to perform the *evaluations and comparisons*. Given the general assumption that vehicle automation technology can be used or developed to benefit existing heavy-vehicle operations or to create new modes of operations, this research was motivated to search for, identify and quantify the value-added of heavy-vehicle automation.

Several operating concepts describing how automation technologies may help improve people throughput have been proposed. They range from an older concept of bus automated highway system (Bus AHS), i.e., a physically segregated and protected system of highway lane(s) dedicated to the exclusive use by fully-automated buses, to a newer concept of automated bus system (ABUS), i.e., a system of closely-spaced automated bus convoys mimicking the light-rail operations on dedicated right-of-way with at-grade crossings along a busy commute corridor of city streets [Tsao and Botha, 2001; Tsao and Botha, 2003]. However, they were developed primarily for the purposes of exploring the technological feasibility and identifying vehicle and infrastructure functional requirements. Therefore, they do not contain sufficient details about system operations; this lack of such details also prevents evaluation of their service quality and operational efficiency.

1.2 Purposes and Scope of the Paper

Although the ABUS concept was motivated to combine the advantages of light-rail operations with the flexibility of the bus operations, its mainline operations mimic their light-rail counterparts, which are not efficient. Also, much synergy in efficiency exists between the mainline operations and local collection and distribution of passengers. Based on the concept of ABUS, this paper proposes a new operating concept that further capitalizes on the flexibility of bus operations. Moreover, the new concept is more complete in that it addresses routing, convoy sizing and scheduling. It provides sufficient details needed for evaluating several key aspects of service quality and operational efficiency.

This paper also compares the performance of the proposed system with the corresponding light-tail and conventional bus systems. The focus is on the more tangible aspects of the comparison. In particular, we compare only the mainline operations and do not speculate on the possible amount of ridership increase resulting from the more flexible service.

Also, we consider the more concrete performance measures like labor requirement, equipment requirement, passenger travel time and fuel saving, but do not address the more illusive performance measures like the environmental impact.

A basic problem with light-rail operations is the issue of low demand-dependence resulting from the difficulty in dealing with spatial and temporal demand variability. A basic problem with any bus system, automated or conventional, is the transfer issue associated with the need for connection of bus routes in order to go from any arbitrary point to another. The new concept alleviates the low-demand-dependence issue of light-rail operations and the transfer issue of a bus system. We first address the low-demand-dependence issue of light-rail operation and then the transfer issue.

Due to the track requirement and the bulky and inflexible mechanical connection between light-rail cars, typical light-rail operations involve fixed-length train operations from one end to the other and back. The capacity provided by such operations is independent of the spatial variability of demand along the mainline; it often is insensitive or independent of the temporal variability of demand. This low-demand-dependence issue leads to operational inefficiency.

It is well known that the relative flexibility offered by the use of buses versus the use of light-rail trains includes the use of the same "dual-mode" buses for mainline operations as well as for passenger collection and distribution off the mainline. However, the degree of net benefit hinges upon travel demand and economy of scale. In theory, one or more routes can be offered to meet the demand between any two points off the mainline (and all points along the route(s)). The population density, the activity patterns and the resulting travel demand between such pairs may be low, and, therefore, some hub-and-spoke structure may be required to lower the cost of providing connection from one point to another. However, such operations require transfers. Such transfers tend to increase travel time and reduce travel time reliability significantly, and tend to expose the passengers to the elements and other possible unpleasant situations. This transfer issue may reduce the service quality significantly.

Although the low-demand-dependence issue and the transfer issue are related, we deal with them separately. Also, we limit the scope of this research to the mainline. The low-demand-dependence issue can be addressed with such a focus. The transfer issue for a bus system including off-mainline operations involves a larger context extending beyond the mainline. However, the way the new operating concept works on the mainline can easily reveal how it works in a bus system involving off-mainline operations.

1.3 Essence of the New Operating Concept and Major Advantages

The ABUS concept *mimics the train-oriented operations of an urban light-rail system along a commute corridor*, with an electronically-linked bus convoy traveling as one unit and with one driver per bus convoy. Therefore, it also suffers from the inefficiency suffered by such urban light-rail operations. Due to the tracking requirement and the operational complexity of mechanical coupling associated with assembling or disassembling a light-rail train, the *size of a light-rail train*, i.e., the number of light-rail cars constituting a light-rail train, tends to be insensitive to the *spatial variability of demand* along the route and may even be constant regardless of this variability. Moreover, although the frequency of light-rail trains can be easily adjusted according to the temporal variability of demand, the train size tends also to be insensitive to the *temporal variability of demand*.

A key advantage of the new bus-automation operating concept, especially when compared to the conventional light-rail operations or ABUS operations in terms of operational efficiency, is that the service can be supplied according to the demand with ease. For example, if a small portion of the mainline enjoys a high demand, more buses can be electronically attached to a bus convoy to satisfy the demand, but the long convoy travels only the portion of high demand and not the entire corridor. This advantage helps alleviate the low-demand-dependence issue. In this concept a bus may travel only part of the mainline, and we refer to the corresponding route a partial route.

Note that this partial route may be a service connecting two stations on the mainline or may be a portion of a route connecting two off-mainline locations. In this paper, we will use the term partial route to refer to the former unless otherwise specified. The way this concept works for the latter can be easily inferred.

A drawback of having partial routes when compared to conventional light-rail operations is that a bus may not travel the whole corridor and hence some passengers of a bus may need to transfer to another bus somewhere and somehow so as to reach the desired destination on the corridor. Such transfers take time, and transferring passengers are exposed to the elements, possible crimes and other unpleasant situations. This drawback is overcome by the new operating concept as follows. With one shuttle bus traveling from end to end and with every other bus *having to* join a shuttle bus (to form a bus convoy on the mainline or to join such a bus convoy already formed on the mainline) *as a condition for using* the dedicated right-of-way, passengers destined for any station long the corridor can reach the desired station through intra-convoy transfer(s), and such an intra-convoy transfer requires a change of bus within the same bus convoy at a station and hence requires no waiting at all. Note that in this operating scenario, buses except the end-to-end shuttle buses travel only part of the corridor. We refer to this operating concept as "Automated Bus System (ABUS) with shuttle-centered convoying and intra-convoy transfer." Since the feature of intra-convoy transfer of this operating concept is enabled by the feature of shuttle-centered convoying, we refer to this new operating concept as "shuttle-centered" ABUS for ease of discussion.

One might argue that such operations or some similar operations might be performed with conventional buses. However, bus-automation technologies enable this shuttle-centered ABUS and contribute to its higher desirability with respect to the corresponding conventional bus operations in at least one key way, among others. The short distance between two consecutive buses in a bus convoy enabled by the technologies shortens the total length of a bus convoy and hence makes the supervision of the operations of a bus convoy by only one driver at the front a possibility, under the *assumption* that buses can be safely linked electronically for travel in a convoy at low to medium speeds along a dedicated right-of-way and through at-grade intersections. Such a possibility offers the potential of drastic reduction in labor cost.

Although the new operating concept is proposed with a focus on mainline operations, it can be integrated with off-mainline operations too. It can not only offer the same mainline benefits, with "dual-mode" buses capable of conventional manual operations and automated operations, but also provide synergy opportunities for combined mainline and local operations.

1.4 Organization of the Paper

This paper is organized as follows. Section 2 summarizes the Phase-I efforts of Tsao and Botha (2003) in duplicating light-rail operations with automated buses, in the bigger context of a four-step deployment sequence. Section 3 proposes the new operating concept. Section 4 defines the scope of evaluation and comparison, including aspects of system to be evaluated, alternatives to be compared, and a reference light-rail system along a commute corridor as a realistic context for evaluation and comparison. Section 5 addresses evaluation and comparison methodology. Section 6 introduces the computer programs and tools developed for the comparisons. Section 7 discusses the numerical results. Concluding remarks are given in Section 8. The reference light-rail system based on which the evaluations and comparisons were conducted is summarized in Appendix A. The functional requirements for the algorithms used for the evaluations and comparisons are given in Appendix B while the computer programs implementing the requirements are given in Appendix C.

2. DUPLICATING LIGHT-RAIL OPERATIONS BY AUTOMATED BUSES

Tsao and Botha (2001) proposed a set of general features for the operations of a automated bus system (ABUS). Their primary focus was to argue that the operations of a conventional light-rail system equipped with dedicated right-of-way, except at at-grade intersections, in a corridor of city streets can be *reproduced* by convoys of automated and electronically linked buses with a driver in the lead bus of every convoy, but with the additional benefit of using the same dual-mode buses for local collection and distribution off the mainline. A set of deployment steps toward this system were proposed in Tsao and Botha (2003), beginning with the use of conventional buses on dedicated right of way except at at-grade intersections. There, they focused on how automation can improve the operations of a conventional busway system equipped with dedicated right-of-way in a corridor of city streets or can even to enable urban busway operations on narrow right-of-way. In both efforts, they focused exclusively on *duplicating* the light-rail services with automated buses, and the primary source of benefit is the ability of the buses to collect and distribute passengers off the mainline. Duplicating light-rail operations with buses requires not only some existing advanced automation concepts like automated lane-keeping and closely-spaced bus convoying but also some new ones, e.g., automated precision turning (into a narrow lane). Closely-spaced bus convoying, if proven safe, enables the lead-bus driver to control the whole convoy and hence leads to reduction of labor cost, when compared to the conventional operations. Note that unlike the original motivation of drastic increase of capacity on a dedicated bus lane for closely-spaced bus convoying (potentially beneficial for very busy but extremely rare situations in the U.S. like the dedicated bus lane in the Lincoln Tunnel connecting the New York City to northeastern New Jersey), the motivation for *closely-spaced bus convoying* as a significant part of ABUS operations *is to reduce labor cost, to reduce disturbance* to surrounding traffic and to avoid possible safety hazards due to the presence of significant gaps between two consecutive buses of a convoy. Of course, the flexibility of using dual-mode buses for both mainline and local operations continues to be a primary motivation for ABUS.

In this paper, we further develop the set of general features into a more complete operating concept, including *bus routing, convoy sizing and scheduling***.** We first summarize the general automation features required for duplicating light-rail operations and then briefly describe the *four-step deployment sequence*.

2.1 A Brief Summary of Bus-Automation Features Required for Duplicating Light-rail Operations

An ABUS operates on right-of-way that is commonly required of a light-rail system and serves the line-haul function. The operations of such a bus-AHS can be thought of as a light-rail system where rail cars are replaced by (self-propelled) buses and physical linkages are replaced by electronic linkages. Buses form closely spaced convoys, mimicking a short train of light-rail cars, so as to minimize disturbance to traffic on surrounding city streets. The buses serve not only line-haul sections but also collect and distribute passengers off the bus-AHS. The flexibility offered by buses extends beyond local passenger collection and distribution. For example, passenger transfer between convoys serving different routes is made possible at a station; a bus can merge with or break off from a convoy at an access or egress location or at a station. Also, passenger transfer can be made within a convoy, which may consist of buses serving different routes. Driverless operations of the trailing buses may significantly reduce labor cost.

2.2 The Four Deployment Steps

The ultimate goal of the four-step deployment sequence is to combine the strengths of a light-rail system and those of a bus system and to formulate new system concepts that offer new and/or better services. At the end of the four steps, the bus operations on the line-haul section mimic those of a light-rail system, but possess much flexibility.

To facilitate understanding, imagine that an existing light-rail system is to be replaced by bus operations. The same four-step sequence can be considered as an alternative to a light-rail option when alternative options are being considered for improving transportation facilities along a busy commute corridor. The sequence can also be viewed as an extension to existing Bus Rapid Transit (BRT) implementations. The operating concepts apply equally well to a network of busways.

It turns out that the availability of right-of-way plays a pivotal role in deploying ABUS systems. In urban settings where the right-of-way allocated to a light-rail system is considered too narrow for human drivers to safely keep the bus in lane at the light-rail speeds, some degree of bus automation may be required. In others, manual driving will suffice. This difference will result in different deployment sequences. However, since the resulting difference in deployment sequencing occurs primarily in Step 1, we will explicitly describe two different sets of possible features for Step 1, rather than specifying two separate four-step sequences.

By the very nature of the setting in which such ABUS systems are designed for, it is likely that the available right-of-way will be narrow for most possible urban implementations. It is logical to describe the operating concept for such a setting first and then discuss the more unrealistic setting. However, since the operating concepts for an ABUS with sufficient amount of right-of-way are simpler, we choose to describe them first and then describe the more complicated operating concepts for an ABUS with narrow right-of-way.

The four steps are summarized below. Implementation issues are addressed in detail n Tsao and Botha (2003).

<u>Step 1: A Conventional Urban Unprotected Busway</u>

We first discuss the urban settings in which the light-rail right-of-way is sufficient for implementing a conventional busway, on which buses can be driven safely by manual drivers at the light-rail speeds, and then those settings in which the light-rail right-of-way is too narrow for manual bus drivers.

Replace light-rail cars with conventional buses, but use the same right-of-way as a conventional busway. Buses travel on the light-rail right-of-way at light-rail speeds, i.e., low speeds in downtown and moderate speeds along urban boulevards, but also collect and distribute passengers in city streets or neighborhoods off the right-of-way.

Movements of buses on the busway are not coordinated, and, as a result, they tend to be more “scattered” along the busway than their light-rail-car counterparts, which are linked mechanically into small trains of light-rail cars. Since signals along the busway are prioritized for bus movement, the scattering of the buses would create a higher degree of disturbance to the surrounding traffic. This is a distinct possibility when demand for travel on the busway increases. This higher degree of disturbance could be so undesirable that the next step would be justified.

We now describe an alternative to Step 1.

<u>Step 1’: An Urban Unprotected Busway with Automated Lane Keeping and Automated Precision Turning/Automated Precision Lane-changing</u>

Because the right-of-way is too narrow for travel by conventional buses, automating the task of keeping buses on the busway may be required. This feature has been referred to as *automated lane-keeping* in the literature. It is possible that automated lane-keeping can be performed only with both lateral control and longitudinal control of the vehicle automated. Other features can be included too, e.g., (*automated*) *precision docking*. Note that precision docking may require both lateral and longitudinal control of the bus.

In addition, buses will need to make turning movements at intersections when they enter or depart the busway. Note that, for entering the busway, they would start the turning movement at the right-hand lane and must enter the narrow busway within as short a distance as possible and without infringing on the right-of-way for the traffic on the adjacent lane on the left. If the buses are to be manually driven onto or off the busway from the crossing street, such turning movements may require a significant amount of additional driving skill or additional right-of-way at the intersections than what a light-rail system would require. (Note that no such turning movements are required for a light-rail system except for those intersections where the light-rail track turns.) Automated turning movement has the potential of minimizing the amount of additional right-of-way required. We refer to this feature as (*automated*) *precision turning*. Note that automated precision turning may require both lateral and longitudinal control of the bus. An alternative to the wide turning movements for entering or departing the light-rail right-of-way is to have buses enter from or depart to the regular traffic lane adjacent to the light-rail right-of-way through a lane-change maneuver. The narrow right-of-way may require automated precision lane-changing.

With the automated lateral and longitudinal control of the bus implemented for purposes of automated lane-keeping, automated precision turning and possibly precision docking, following the bus ahead at a safe distance can be automated without much difficulty. This feature has been referred to *automated vehicle following* in the literature.

As in Step 1, movements of buses on the busway are not coordinated, and the degree of disturbance to the surrounding traffic could be so undesirable that the next step would be justified.

<u>Step 2: A Busway with Manual Bus Convoying through ITS Technologies</u>

The main goal is to reduce the degree of disturbance to the surrounding traffic. Through the use of ITS technologies, including communications and fleet management technologies, the movements of buses on the busway can be well coordinated so that they form bus convoys and bus convoys are properly spaced. Such coordination may reduce the disturbance to surrounding traffic. We refer to this feature as *manual bus convoying*.

As the demand for travel on the busway further increases, such coordination may still incur an unacceptable amount of disturbance to the surrounding traffic. The next step is designed to remedy the situation.

The essence of this step is to mimic light-rail operations with ITS buses, except the presence of a driver for each and every bus of a convoy.

<u>Step 3: Automated Closely-spaced Convoying of buses to further reduce the disturbance to the surrounding traffic. (An Automated Busway)</u>

Further shorten the following distance between two buses with the help of automation, and organize buses into short closely spaced convoys. Safety and ride quality are also important performance measures. This step features "feet-off" driving for the trailing buses, and we refer to this feature as *automated closely-spaced convoying*.

If features of automated lane-keeping, automated precision docking and automated precision turning have not been implemented before this step, then they can be implemented as part of this step. Although they can be implemented in a separate future step, we assume that this is implemented in this step, if they have not already been implemented. With their implementation, the required right-of-way can be reduced, and ride quality improved. With their implementation, driving on the line-haul section becomes "hands-off" for both lead and trailing buses.

Note that this motivation for automated closely spaced convoying is completely different from the motivation for platooning, which is to double or triple the capacity of an automobile AHS by packing a freeway lane with automobiles safely.

Although the task of longitudinal control of a lead bus may also be automated, the driver of the lead bus is responsible for anticipating intruding vehicular or passenger traffic from the surrounding roadways or sidewalks into the right-of-way and reacting to such and other non-nominal events by overriding automated driving.

<u>Step 4: Driverless operations for trailing buses in a convoy. (Another ABUS Concept)</u>

If safety permits, the trailing buses of a convoy may not require a driver. However, the lead bus of a convoy continues to require a driver. Note that if the driverless operations cannot be made sufficiently safe or efficient, the deployment can stop at the previous step, which is by itself an automated busway system.

This step makes the bus operations on the line-haul section resemble the current light-rail operations. This step has the potential of significantly reducing the labor cost for operating the system described in Step 3. We refer to this step as *Driverless bus following*. Due the absence of rail tracks and drivers, steering failure on the part of such trailing buses may cause significant safety hazards. This issue must be studied thoroughly.

*The essence of this step is to mimic light-rail operations with automated buses***.**

2.3 A Summary of the Four Steps

These four steps as well as their main features, main benefits and main traffic issues are summarized in Table 1. Note that, among other issues, we focus on only the traffic issues because minimizing disturbance to the surrounding traffic is a major driving force behind the deployment sequence.

Table 1: A Four-step Deployment Sequence for the ABUS system

Step	Main Features	Main Benefits	Main Traffic Issues
1 Conventional Busway (with sufficient right-of-way)	• Bus operations on light-rail right-of-way	• Same bus for line-haul and collection/distribution	• Scattered buses disturbing surrounding traffic
1' Busway with Automation to Enable Bus Operations on Narrow Right-of-way	• Automated precision turning/automated precision lane-changing • Automated lane-keeping • Automated bus-following • Automated precision docking (if desired)	• Same bus for line-haul and collection/distribution • Automation enabling busway operations on narrow right-of-way	• Scattered buses disturbing surrounding traffic
2 Manual Bus Convoying Through ITS	• Clustering buses to reduce disturbance via ITS technologies	• Reduction of disturbance to surrounding traffic due to signal preemption or priority	• Possible excessive disturbance as demand grows
3 Automated Closely-spaced Convoying	• Automated closely-spaced convoying • Automated precision turning/automated precision lane-changing, automated lane-keeping, automated vehicle-following and precision docking if not already implemented	• Reduction of disturbance to surrounding traffic due to signal preemption or priority	• Safety
4 Driverless Bus-following	• Absence of driver on trailing buses	• Reduction of labor cost	• Safety

In the next section, we further develop the general features of an ABUS into a *new* and more *complete* operating concept**.**

3. A NEW AND MORE COMPLETE OPERATING CONCEPT FOR AN AUTOMATED BUS SYSTEM ON A CORRIDOR OF CITY STREETS

Now that how ABUS operations can duplicate the light-rail operations on dedicated right-of-way along a corridor of city streets has been described, we further develop a new and more complete operating concept that more fully capitalizes on the flexibility of bus operations and the technology of ABUS and can outperform light-rail operations in operational efficiency (on the mainline).

The efficiency potential of the new concept has two sources: flexibility of bus operations and automation technology. In order to clarify operations and the corresponding efficiency gains enabled by the flexibility of bus operations from those enabled by automation technologies, we will separate the description of the new concept into two disjoint parts: flexibility features and automation features.

In describing the flexibility features, we consider situations in which the least amount of automation technology is required. For example, the right-of-way is sufficiently wide so that automated lane keeping as well as other automation technologies like automated precision turning are not required. Also, labor cost reduction and minimization of disturbance to surrounding traffic enabled by automated closely-spaced convoying are not required.

This separation is consistent with the discrete steps of the four-step deployment sequence. More precisely, the flexibility features require no automation technologies and expand the Step 2 - Manual Convoying Through ITS Technologies - into a new and more complete operating concept. Moreover, the flexibility features plus a new automation feature not addressed in Step 4 expand the Step 4 - Driverless Bus-Following - into a new and more complete operating concept.

3.1 Flexibility Features and Their Motivation

Travel demand for public transportation along the mainline of a commute corridor is clearly variable. The variability comes in at least two forms: spatial variability and temporal variability. The spatial variability refers to the different number of customers riding on different sections of the mainline, which results from the diverse travel origins and travel destinations of the customers. The temporal variability refers to the different number of customers riding on the mainline in different time periods.

Due to the tracking requirement and the operational complexity of mechanical coupling-decoupling associated with assembling or disassembling a light-rail train, the *size of a light-rail train*, i.e., the number of light-rail cars constituting a light-rail train, tends to be insensitive to the *spatial variability of demand* along the route and may even be constant regardless of the variability. Moreover, although the frequency of light-rail trains can be easily adjusted according to the temporal variability of demand, the train size tends also to be insensitive to the *temporal variability of demand*. These operational inefficiencies can be avoided with the flexible use of buses.

To facilitate understanding of the features, we assume that the Step 2 - Manual Bus Convoying Through ITS Technologies - has been implemented.

Note again that this partial route may be a service connecting two stations on the mainline or may be a portion of a route connecting two off-mainline locations. In this paper, we use the term partial route to refer to the former unless otherwise specified. However, the way this concept works for the latter can be easily inferred.

Feature 1 - End-to-end Shuttle with a Driver Allocated

This feature provides a shuttle bus traveling the entire corridor from one end to the other. One driver is allocated for each shuttle. As will become clear later, the driver may not stay on the shuttle exclusively while traveling on the mainline. Rather, it will stay with the lead bus of the convoy containing the shuttle. The driver is needed to supervise the operations of a closely-spaced automated bus convoy for safety.

The shuttle bus is dispatched according to a fixed schedule. This shuttle service ensures that a passenger can ride the bus to reach any station from any other station as long as there is capacity to accommodate the passenger. If, when and where demand warrants, one or more uses can be electronically attached to the shuttle bus to form an end-to-end convoy shuttle.

We at times will refer to this end-to-end automated shuttle bus along the whole mainline simply as the shuttle. Note that this shuttle bus route may be a portion of another longer route connecting two off-mainline points in the corridor.

Feature 2 - Partial Mainline Coverage as Part of an Off-Mainline Bus Route

This flexibility feature is to provide partial routes, each of which may connect any two stations along the mainline.

Feature 3 – Scheduling of All Partial Routes In Synchronization with the Shuttle

Buses of all routes are scheduled in such a way that convoy composition of any section remains identical throughout the day. This can be implemented with a fixed headway for all routes or one fixed headway for a period of time. This feature is motivated to simplify the routing and scheduling so as to avoid confusing the passengers. (This feature can be relaxed if appropriate.)

When off-mainline routes containing a partial route on the mainline are involved, the buses may have difficulty in keeping the schedule. Such disturbances may cause some inconvenience and even confusion to the passenger. See Feature 9 for further discussion. Until then, we assume adherence to schedule.

Feature 4 – Sufficient Capacity for Coverage of Individual Sections

The combined capacity of all the partial routes should meet the section demand for all sections. The determination of the actual partial routes should depend on the demand at individual sections of course. Any such partial route may also be a portion of a longer route connecting two off-mainline points in the corridor.

Feature 5 - Shuttle-Centered Convoying

Each and every bus traveling on the mainline must be part of a convoy that contains an end-to-end shuttle. In other words, each and every convoy must have a shuttle in it; as the shuttle travels from one end to the other, buses serving the partial routes join a convoy at their entry points and separate from a convoy at their destinations. An entering bus may first enter a station and then wait for a convoy to join or may join from behind a convoy arriving at or having already arrived at a station.

Feature 6 - Driver Presence of the Lead Bus of a Convoy, Not Necessarily the Shuttle

For safety purposes, a driver is required for every convoy. Since a driver is allocated to every shuttle, by Feature 5, there will be one driver per convoy. The driver will stay with the lead bus of the convoy, rather than always staying with the shuttle itself. Recall that a bus may join a bus convoy "from the front" by first entering a station on the mainline from an off-mainline street, then waiting for the approaching convoy, and finally becoming the lead bus of the combined convoy. In such a case, the driver of the approaching bus convoy switches to the new lead bus.

Feature 7 – No-wait Intra-Convoy Transfer on the Mainline

Since all buses when traveling on the mainline travel with a shuttle, transferring from a bus of one partial route to a bus of another partial route within the same convoy can be conducted without wait at all. In such cases, a passenger needing to transfer simply gets off the bus of one route and immediately gets onto the bus of another route of the same convoy. This is another direct consequence of Feature 5.

Feature 8 – No-wait (Intra-convoy) Transfer(s) Throughout the Travel on Mainline

Regardless of the destination of a passenger, if she chooses to board the shuttle, she can reach any station without having to transfer during the ride. Therefore, she would prefer boarding the shuttle unless she has a closer destination and another (partial-route) bus in the convoy also goes to the destination. In any case, there would be a preferred bus of the convoy for her. It is possible that there is no capacity (either as a seat or as standing room) on the preferred bus when she boards the convoy. In such a case, she can board one of the other buses in the convoy and reach the destination through transfer(s). Recall that buses of all routes are scheduled in such a way that convoy composition of any section remains identical throughout the day, i.e., Feature 3. Also recall that the routing of the shuttle and the partial routes and their scheduling is performed in such a way that the demand at all sections is satisfied, i.e., Feature 4. There is room for every passenger to reach her destination regardless of her origin and destination. Therefore, she can go from any mainline station to any other with no-wait intra-convoy transfer(s)

When partial routes are portions of off-mainline routes, the shuttle may no longer be the preferred route for all passengers because some passengers may be destined for an end point off-mainline that is served by one of the partial routes. Also, in this case, a passenger may enter the mainline on a bus of an off-mainline route serving two off-mainline end points. If her destination is another off-mainline end point that is served by another off-mainline route, then she would prefer to transfer to the bus serving her destination, instead of the shuttle.

Given Feature 3, transfer on the mainline between any two partial routes with overlapping mainline sections involves no wait. Moreover, transfer on the mainline between two buses serving two non-overlapping partial routes can be achieved by one transfer from one of the two routes to the shuttle followed by another transfer from the shuttle to the other route. None of these two transfers involve any wait. As a result, transfer on the mainline between any two partial routes involves no wait.

This reasoning can be extended to cases where (mainline) partial routes are portions of longer off-mainline routes. It can be shown that transfer on the mainline between any two off-mainline routes (containing mainline partial routes) involves no wait.

Feature 9 - Synchronized Schedules for On-bus Waiting (as Opposed to On-street Waiting):

The operations of off-mainline bus routes can be synchronized with the shuttle operations so that such a bus would arrive at the mainline at the arrival time of a convoy and the connection would entail minimum wait on the part of the entering bus or the convoy. Again, the entering bus can wait in a station if it arrives at the station before a convoy, rather than waiting at a nearby location. The synchronization performance is subject to the stochastic nature of the local traffic condition and demand, and must be studied accordingly. However, the waiting is done in a bus instead of on the street and even in the elements or other unpleasant situations. A simple way of operating multiple off-mainline bus routes with overlapping mainline coverage is to partition a day into several time periods and, for each period, run one bus for each and every off-mainline bus route for a fixed amount of time, e.g., every 10 minutes so that the scheduled arrival time of a off-mainline bus at a station (either for entering into or departing from the station) coincides with the arrival time of a convoy.

3.2 Automation Features and Their Motivation

Feature 10 - Reduction of Right-of-Way Requirement

As discussed earlier, automation may reduce the amount of right-of-way required for bus operations on a corridor of a metropolitan area and hence may actually enable the implementation of such a bus system.

Feature 11 - Driveless Bus Following

Perhaps more importantly, automation may safely enable driverless bus-following for closely-spaced bus convoying. More precisely, the short distance between two consecutive buses minimizes the total length of a bus convoy, and hence may enable safe supervision of a bus convoy by the driver, who supervises the operations at the front of the lead bus. This will significantly reduce the labor cost required to support the flexibility features just described.

Feature 12 - Less Disturbance to Surrounding Traffic

The shorter bus following distances may reduce the amount of disturbance to the surrounding traffic.

Feature 13 - Efficient Synchronization of Mainline Operations and More Efficient Coordination between Mainline and Local Operations

Automation may also significantly improve the degree of synchronization between traffic signaling on the mainline and the degree of coordination between the mainline operations and local bus operations, particularly when combined with transit signal priority on the mainline and along the off-mainline routes.

This last feature is not part of Step 4. but can be viewed as an extension to Feature 9 of the Flexibility Features.

The ABUS operating concept is captured by the Step 4 of the four-step deployment sequence described in Section 2. The new operating concept of Shuttle-Centered ABUS can be viewed as the ABUS system plus all the Flexibility Features, i.e., Features 1 through 9, and (Automation) Feature 13. Or, equivalently, it can be viewed simply as Step 4 plus the Flexibility Features and (Automation) Feature 13.

4. SCOPE OF EVALUATION AND COMPARISON AND A REFERENCE CORRIDOR

As discussed earlier, for evaluations and comparisons, we focus on only the mainline operations. We briefly describe the alternatives for comparison, the aspects for comparison and a reference corridor.

4.1 Alternatives for Comparison

We compare the new concept with both the light-rail operations and the conventional bus operations. We focus on the operational efficiency. Other aspects of the comparisons exist; they include capital cost, environmental impact, ride quality. They are worthy subjects for further study.

The scope of comparison include the following three operating concepts:

- Light-rail
- Shuttle-centered Busway: Conventional Busway or Step 2 (i.e., mimicking light-rail operations with ITS buses) + Flexibility Features
- Shuttle-centered ABUS: ABUS or Step 4 (i.e., mimicking light-rail operations with automated buses) + Flexibility Features

4.2 Aspects for Comparison

When compared to the light-rail operations, relative efficiency of Shuttle-centered ABUS operations has at least two major sources: mainline operations alone and local portion of integrated mainline-local operations. We focus on the mainline operations in this paper. The goal is to show that the potential of the new operating concept for improving the mainline operations alone is substantial; the overall potential when the local portion is also considered should be even higher.

We focus on the equipment requirement, labor requirement, fuel consumption and total passenger travel time.

4.3 A Reference Corridor

To make the evaluations and comparisons more realistic, we use the Santa Clara Light-rail System operated by the Valley Transportation Authority (VTA) of the Santa Clara County, California as a reference corridor and a reference light-rail system. See Appendix A.1 for a system map.

At the time of data collection for this research, the system had three lines: Mountain View to I880/Milpitas, Baypointe to Santa Teresa, and Ohlone/Chynoweth to Almaden. See Appendix A.1 for the route map. (Since then, the first line, i.e., the Mountain View to I880/Milpitas line, has been extended significantly.) Note that three lines essentially constitute one commute corridor and can be viewed effectively as one liner. On the first line (Mountain View to I880/Milpitas), there are only two stations beyond the Baypointe station (toward the East), which is the transfer station between the first two lines (i.e., the Mountain View to I880/Milpitas Line and the Baypointe to Santa Teresa Line). In addition, there are only two stations on the third line (i.e., the Ohlone/Chynoweth to Almaden Line) beyond Ohlone/Chynoweth, which is on the second line.

We focus on the portion of the first line connecting Mountain View to Baypointe and the entirety of the second line (connecting Baypointe to Santa Teresa) and treat the combination as one line serving the commute corridor from Mountain View to Santa Teresa. The passenger volumes associated with the two stations outside of this focus have been added to the actual passenger volumes associated with the Baypointe Station as if those passengers destined for the two ignored stations were destined for Baypointe instead. We neglected the third line, i.e., the Ohlone/Chynoweth to Almaden Line altogether. More importantly, we assume that there is one single line connecting Mountain View directly to Santa Teresa, instead of treating the actual case of having two separate but connected lines. These assumptions are made because the focus of this study is on a single commute corridor served by one light-rail line. Due to the differences between the actual system and the assumed single line, the Santa Clara Light-rail System and the corridor it served constitute only a reference model.

This reference model is described in more detail in Appendix A. As mentioned earlier, Appendix A.1 is the route map for the Santa Clara Light-rail System as of March 2003. Appendix A.2 contains the volume data along the line hypothetical line, which were derived from the volume data of the system provided by the Valley Transportation Authority. Appendix A.3 contains the distances between pairs of consecutive stations on the hypothetical line.

The passenger data that are available to us are the on and off data for the north-to-south direction only (from Mountain View to Santa Teresa). We study only one direction. However, the performance of the three alternatives should be the same or at least similar for the other direction due to symmetry. We use the headway information published in the light-rail schedule in our evaluations and comparisons.

No origin-destination trip numbers were available, and, therefore, we had to estimate them. Details about this estimation process will be discussed in more detail in the next section.

5. EVALUATION AND COMPARISON METHODOLOGY

We discuss the assumptions, dimensions for evaluation and comparison, performance measures and their calculations, organization of evaluation and comparison results and origin-destination trip volume estimation.

5.1 Assumptions

- Common size of a light-rail car and a bus, either conventional or automated: We assume that a light-rail car and a bus, either conventional or automated, have the same capacity. We also assume that the capacity is 50 per light-rail car or bus.
- Common headway for light-rail trains and the shuttle bus from one end to the other.

- Common headway for all bus routes, including the shuttle route (from one end to the other) and all the partial routes. This is assumed for both automated bus convoying or non-automated bus convoying.
- Every light-rail train or a bus convoy stops at every station.
- Common travel time for light-rail cars and buses. Estimated distances and travel times between every pair of consecutive stations are given. These estimated travel times will be used in calculating performance measures related to equipment and labor requirements.
- Although the third (fuel consumption) depends on the number of passengers, we assumed that the fuel consumption is insensitive to the load in this study. We also assume a 10% reduction of fuel consumption for automated closely-spaced bus convoying. (Ulmer, 1999).

5.2 Dimensions for Evaluation and Comparison

The following dimensions are considered in developing and specifying evaluation and comparison cases:

- **Time periods**: The Santa Clara light-rail system partitions its operation hours into four time periods: AM peak (5:30 – 8:30), mid-day peak (8:30– 14:30), PM peak (14:30-17:30), and off-peak (17:30 – 5:30). Although the current demand pattern depends on the time period, both the headway and length of light-rail trains seem approximately constant within the first three of the four time periods, i.e., from 5:30 through 17:30. In fact, the headway is exactly10 minutes from 6:05 through 18:35. In the rest of this paper, we focus only on the first three periods and will assume such constancy for three periods. More precisely, the headway is 10 minutes, and the train length is 4 cars. The off-peak period is not as interesting because of the low demand. Having said that, we note that the potential saving in equipment may be very significant. Rather than using a train of two articulated light-rail cars, a single shuttle route between the two ends suffices. In other words, a single bus, without any partial-route buses, typically suffices for the low demand, and therefore the equipment requirement can be cut in half by using buses instead of light-rail trains.
- **Demand**: The current demand pattern depends on the time period. (See Appendix A.2.) To anticipate demand growth and to evaluate and compare the alternatives at higher demand levels, we let the demand vary. In addition to the studying the performance under the current demand, we study the performance under inflated demand. Specifically, we inflate the trip volume by increments of 25% (i.e., to 125%, 150% … of the current demand) for every origin-destination pair until the capacity of the current light-rail or the capacity of its bus-convoy alternatives (designed to meet the current light-rail demand) is exceeded by the inflated demand. (Inflating the demand to a higher volume would necessitate change of the headway or frequency of a light-rail train or bus convoy, if the demand is to be satisfied. Since many factors are involved in headway determination, changing headway is beyond the scope of this paper.)

- **Number of partial routes and the routes themselves**: We considered cases involving two partial routes or one partial route. Note that all these partial routes and the shuttle route have the same headway. (The headway is 10 minutes, as discussed earlier.) Also note that these partial routes are not obtained through any rigorous optimization process. Such optimization must consider passenger demand for both directions, but the data for the opposite direction are not available to us. Moreover, routes are determined based on a number of non-quantitative factors. Moreover, such optimization requires full consideration of the stochastic nature of the passenger demand, and computer simulation is a more appropriate tool. Such optimization is beyond the scope of this research, but is worthy future research subject.
    - Case of two partial routes: A longer one contains the shorter one. The longer one connects Fair Oaks (Station 10) in the north and Ohlone-Chynoweth (Station 38) in the south; the shorter one connects Great America (Station 14) and Discovery Museum (Station 32).
    - Case of one long partial route: The only partial route is exactly the longer partial route of the previous case – from Fair Oaks (Station 10) in the north to Ohlone-Chynoweth (Station 38) in the south.
    - Case of one short partial route: The only partial route is exactly the shorter partial route of the two-partial route case – from Great America (Station 14) to Discovery Museum (Station 32).
    - Case of one partial route on the south half: The only partial route connects Orchard (Station 20) and Santa Teresa (Station 42).

5.3 Performance Measures and Their Calculation

We use four performance measures. They are defined as follows.

- **Total equipment time**: The unit of this quantity is light-rail-car-minute or bus-minute. Within a given period of time, i.e., any of the four different time periods (in which both the headway and train-convoy length are assumed to be constant), this quantity is defined as the total amount of time that is required of one light-rail car or bus to serve all the route(s) according to the schedule. For light-rail operations, it is calculated in the following way. First, figure out the total number of light-rail trains departing from the north end during the time period. Multiplying this number first by the number of light-rail cars in a train and then by the travel time from the north end to the south end. For bus operations, this is the sum of the individual components calculated for all the routes, including the end-to-end (shuttle) route and all the partial routes. Each component is calculated by multiplying the number of schedule runs for the bus route during the time period by the travel time of that route.
- **Total labor time**: The unit of this quantity is driver-minute, either light-rail driver or bus driver. In making comparisons, we make no distinction between the labor rate of the light-rail driver and that of the bus driver. In addition, we make no distinction between the labor rate between the driver of an automated bus and that of a conventional bus. The calculation is similar to that of the total

equipment time, except that one light-rail train or automated bus convoy requires only one driver.

- **Total fuel consumption and % of fuel saving**: The fuel saving refers to the saving of fuel due to reduced air resistance resulting from closely-spaced automated bus following, and hence this measure is applicable to only the case of Shuttle-centered ABUS with respect to the case of Shuttle-centered Busway. The fuel consumption for the two bus-convoy alternatives is assumed to be 10 miles per gallon. The focus is actually on % fuel saving. Simulation results for commercial truck operations (Ulmer, 1999) indicate that both lead vehicle and trailing vehicles can benefit from automated closely-spaced convoying. We borrowed those simulation results for extrapolation to bus operations, and assume that only trailing buses enjoy fuel saving, and the saving is 10%. According to Ulmer (1999), the fuel saving for a trailing truck depends on the speed and the vehicle-following distance, and savings may range between 0% and 20%.
- **Total passenger time**: The unit of this quantity is passenger-minute. The average speed of all three systems is 0.6 miles per minute, i.e., 36 miles per hour. Its calculation is similar to the calculation of total labor time, except that the focus in this case is on the time passengers spend on travel, instead of the drivers. *This quantity is actually the same for all the cases evaluated and compared.*

Transit signal priority is commonly implemented along light-rail mainline, and it may cause disturbance to surrounding traffic. Due to the stochastic nature of the problem, it is studied and reported separately.

5.4 Organization of Evaluation and Comparison Results

Recall that the focus of our evaluation and comparison is on four criteria: the equipment requirement, labor requirement, fuel consumption (for the bus-convoy operations, either conventional or automated) and total passenger travel time. Of the four, the first two depend completely on the route structure and the frequency of service. Although the third (fuel consumption) depends on the number of passengers, we assumed that the fuel consumption is insensitive to the load in this study. Therefore, the first three of the four criteria depend only on route structure and service frequency. The fourth criterion – passenger travel time – depends on the demand. Since we consider only cases where the demand does not exceed the capacity, the passenger travel time actually does not depend on the route structure. Therefore, we will report the results about the first three criteria - equipment requirement, labor requirement, fuel consumption - separately from those for the fourth criterion – passenger travel time.

As mentioned earlier, we inflate the current demand to study how much more demand can the individual route structures can accommodate. The highest amount of demand a route structure can accommodate depends on the time period. The highest amount of demand a route structure can accommodate will also be reported.

The evaluation and comparison results for the four route structures are summarized in Tables 2, 3, 4 and 5 of Section 7. Those regarding the passenger travel times are summarized in Table 6 of Section 7. The highest amounts of demand a route structure can accommodate are summarized in Table 7 of Section 7.

Note that we study only the performance for one direction only (from north to south, i.e., from Mountain View to Santa Teresa). Are these data are available to us; the result for the other direction is expected to be similar.

5.5 Origin-Destination Trip Volume Estimation

As mentioned earlier, the only data available to us are the numbers of passengers getting on or off a south-bound light-rail train at the individual stations, and no origin-destination trip numbers were available, and, therefore, we had to estimate them. We used the method of constrained entropy maximization. In short, entropy can be interpreted as the amount of uncertainty contained in a distribution. The more uncertainty there is in a distribution, the higher the entropy. In a more visual term, the flatter the distribution or, equivalently, the wider spread, the higher the entropy. In this current context, an infinitely many possible origin-destination trip numbers can result in the observed on- and off- numbers. The method of constrained maximum entropy produces the one (out of the infinitely many possibilities that result in the observed on and off counts) that is the most uncertain, is the flattest or, equivalently, has the widest spread.

This technique has been used for estimating the trip volumes associated with different pairs of origin-destination zones in a region (Fang and Tsao, 1995). In that case, the method is closely related to the so-called "gravity model." In this paper, we use it to estimate the trip volumes associated with pairs of origin-destination stations along a light-rail or bus corridor. The method has also been used in studying the collision probability and impact force in the context of AHS safety by Tsao and Hall (1994). The methodology of entropy optimization has been treated recently in Fang, Tsao and Rajasekera (1997).

Constrained entropy maximization involves the generic entropy function and a set of constraints. Our context requires only linear constraints. The only input required for constrained entropy maximization is the constraint set. A C program that produces the constraint set based the on and off volumes at all the stations has been developed and will be discussed in the next section. A set of FORTRAN and C programs that solve the resulting linearly-constrained entropy maximization problem has also been developed and will also be discussed in the next section.

6. COMPUTER TOOLS

A set of computer tools have been developed. They have been developed not only to facilitate this particular study but also to allow similar studies by the reader for corridors of their choice. However, we use our particular use to illustrate their usage.

6.1 Solicitation of Problem Input for Performance Evaluation and Comparison, Given OD Trip Estimates

This C program acts as the user interface to get the data from the user. The user data is checked for any errors or overruns, ex: whether the entered partial route is within the limits of the corridor and so on. As a default behavior, the program itself can also generate partial routes when the user doesn't provide any specifics. This program is shown in Appendix C.1.

- The number of stations the users wishes to have.
- Partial routes entered by the user or default partial routes from program.
- Speed of the bus.
- Values of the headway for different time slots i.e. am, pm, midday and off-peak.

The output of this program is illustrated below (when run in its default mode):

Enter the number of stations: **42**
Do you wish to enter your own partial routes (y/n) [*no*]*:* **no**
Partial routes #1: 7, 21
Partial routes #2: 14, 28
Partial routes #3: 21, 35
The Speed of the Bus is: 1 mile(s)/min
Do you wish to enter your own headway times (y/n) [*no*]*:* **no**
The ampeak, pmpeak, midday and offpeak values are: 10 15 15 20

6.2 Estimation of Trip Origins and Destinations – An Entropy Maximization Program Implementing the Gravity Model for Trip Distribution

Constraint Generator Program:

This C program also has a user interface which is very flexible and lets the user choose the file names of the user input data files (like the ON data file, OFF data file, distance file) and the output files. This feature is helpful when the user is trying to generate constraints for different sets of data simultaneously (or different scenarios like peak time vs. off-peak time; or comparing two different corridors; etc) by avoiding the confusion of overwriting data files. Constraints are formulated based on the data obtained from the light rail system. The formats of the generated data and constraints adhere to the requirements of Fortran solver. The C program is currently designed to handle a maximum station of 100 but can be easily scaled to accommodate any higher number. The program is provided in Appendix C.2.

<u>Input</u>

A discussed earlier, we focus on the north-to-south direction only. In order to define the constraints, the following data was collected from Santa Clara VTA Light Rail.

- Number of stations on the line
- Distance between two consecutive stations
- Traffic, i.e.,passenger counts

There are 42 stations on the line, and 41 sections. For each station, we need:

- Passenger arrival data at each station. See Appendix A.2.
- Passenger departure data from each station. See Appendix A.2.

The traffic is categorized into four different time periods based on the demand uniformity. The above data is collected for each of the four time periods at every station on the corridor. The 4 time periods are:

- AM peak (Time: 5:30 AM to 8:30 AM)
- PM peak (Time: 8:30 AM to 2:30 PM)
- Mid day (Time: 2:30 PM to 5:30 PM)
- Off peak (Time: 5:30 PM to 5:30 AM)

Separate analysis is done for all the each of these data sets. The data obtained from the light rail system is shown in Appendix A.2.

Scope: Origins and destinations refer to the start and end stations of the trips on the mainline, i.e., the origin and destination stations. They do not refer to the locations of the riders' true trip origins or destinations off the mainline.

Input Type	***Input File Name***	**Description**	***Representation***
User	---	The total number of stations in the corridor.	N
File	distance.txt	A single dimension array of distances between each station.	$\{D_{1,2}, D_{2,3}, \ldots D_{N-1,N}\}$
File	on.txt	A single dimension array of number of passengers boarding the light-rail.	$\{I_1, I_2, I_3, \ldots I_N\}$ $I_N = 0$ for end station
File	off.txt	A single dimension array of number of passengers alighting the light-rail.	$\{O_1, O_2, O_3, \ldots O_N\}$ $O_1 = 0$ for start station

Generating the constraints using a C Program

As stated earlier, the VTA data sets are used to generate the constraints and then are solved to obtain the OD volume data.

The program reads the input (both from the user and the files) and generates the constraint matrix according to the requirements of the Fortran program and outputs (or prints) it to a file. The C program has the following modules (or sub-routine functions),

1. Main function:
   Defines all the required variables, arrays & matrices and initializes them. It dispatches control by calling various functions/modules in the program.

2. User Interface and Input verifier:
   Reads the user entered data like the number of stations, input & output filenames, average trip length. It also performs negative checking for incorrectly entered data, like a non-existing input filename or out of bound average trip length or insufficient data entries in the input file etc. A typical output of the user interface of this program is shown below – Sample User Interface of the Constraint Generator Program.

3. File reader:
   Reads the data from the user specified / default files into the computer memory and organizes it to be manipulated by other modules. The input files include the distance file, departure/boarding data (ON) and arrival/alighting data (OFF).

4. Constraint matrix generator:
   Constraints matrix is generated using the number of stations under consideration and the ON and OFF data sets stored in the array variables.

   The constraint matrix generator puts a "1" for every valid parameter and a "0" for every invalid parameter. Consider an example with N = 5 stations, S_1, S_2, S_3, S_4, S_5, with $\{I_1, I_2, I_3, I_4, I_5$, where $I_5 = 0\}$ as in-bound passengers and $\{O_1, O_2, O_3, O_4, O_5$, where $O_1 = 0\}$ as the out-bound passengers at respective stations. The generated constraint matrix looks as follows:

$$\begin{bmatrix} 1&1&1&1&0&0&0&0&0&0\\ 0&0&0&0&1&1&1&0&0&0\\ 0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&0&0&1\\ 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&1&0&0&0&0&0\\ 0&0&1&0&0&1&0&1&0&0\\ 0&0&0&1&0&0&1&0&1&1 \end{bmatrix} \Longleftrightarrow \begin{bmatrix} I_1\\ I_2\\ I_3\\ I_4\\ O_2\\ O_3\\ O_4\\ O_5 \end{bmatrix} \Longleftrightarrow \begin{bmatrix} X_{1,2} + X_{1,3} + X_{1,4} + X_{1,5}\\ X_{2,3} + X_{2,4} + X_{2,5}\\ X_{3,4} + X_{3,5}\\ X_{4,5}\\ X_{1,2}\\ X_{1,3} + X_{2,3}\\ X_{1,4} + X_{2,3} + X_{3,4}\\ X_{1,5} + X_{2,5} + X_{3,5} + X_{4,5} \end{bmatrix}$$

   The ultimate goal is to solve for $X_{1,2}$, $X_{1,3}$, … $X_{N-1,N}$, which constitutes the OD (Origin-Destination) volume data.

5. Distance matrix generator:
   Given the distance array having the distances between consecutive stations, this function generates a matrix having the distance between all possible combinations of stations.

   Given 5 stations with distances $\{D_{1,2}, D_{2,3}, D_{3,4}, D_{4,5}\}$ it generates a 5x5 matrix as follows:

$$\begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} \\ d_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} \end{bmatrix}$$

   Where $d_{1,1}, d_{2,2} \ldots d_{5,5} == 0$.

6. File writer:
   Writes formatted output as needed by the Fortran program to the files specified by the user (or to default file locations).

<u>Output</u>

The Outputs of the C program are as follows:

Output Type	***Output File Name***	**Description**	***Representation***
File	data.a	2 dimension constraint matrix, represented by 1's and 0's. LHS of the equation.	$[\ X_{1,2}, X_{1,3}, \ldots X_{N-1,N}$
File	data.b	1-dimension array of ON and OFF data. RHS of the equation.	$\{I_1, I_2 \ldots I_{N-1}, O_2, O_3 \ldots$ O **Note**: I_N and O_1 are miss
File	data.sizes	Required by the constrained entropy maximization program to determine the number of input data elements.	$\{n, m\}$ where, n = 2*(N-1) and, m = N * (N-1) /
File	data.perturbation	Set the value of μ to 1. This causes the objective function to be a pure entropy function. (The solver is capable of solving entropy maximization problem with an additional linear function in the objective function, in addition to the pure entropy function.)	$\mu = 1$
File	distance.out	2-dimension distance matrix showing the distance between any 2 given stations.	$[\ d_{1,1}, d_{1,2}, \ldots d_{N,N}\ ]$

Sample User Interface of the *Constraint Generator* Program:

The following table shows how to interpret the output of the C programs.

Text Type	**Meaning**
Italic text	Indicates the program display / output.
Bold text	Indicates the user's input to the program.
[Text within brackets]	Indicates the DEFAULT value assumed by the program, incase the user doesn't input any data.
Normal text	Also indicates the output of the program

The user interface is illustrated below.

Enter the number of Stations: **42**
Enter the file_name for Distance between Stations [*distance.txt*]*:*
Enter the input file name for Departure Data [*departure.txt*]*:* **am_peak_on.txt**
Enter the input file name for Arrival Data [*arrival.txt*]*:* **am_peak_off.txt**
Enter the output file name for LHS [*data.a*]*:*
Enter the output file name for RHS [*data.b*]*:*
Enter the output file name for sizes [*data.sizes*]*:*
Enter the average trip length (MIN 0.6 - MAX 25.5) [*13.0*]*:*

INFO: Wrote Distance to output file: distance.out
INFO: Wrote M,N data sizes to output file: data.sizes

Note: The current program does not impose a constraint on the average trip length due to lack of such data. However, if the information is available, the program can be easily extended to include the constraint. Although this user interface program does ask for the average trip length, the information is not used in the constraint-generation program.

Solver Programs for Linearly-Constrained Entropy Maximization :

The constrained entropy maximization method is implemented with a set of Fortran and C programs. This set of solver programs is provided in Appendix C. 3.

The solver uses the constraint generator output files (data.a, data.b, data.sizes, data.perturbation) as input to calculate the OD volume data. OD volume data (the number of passengers traveling from station i to station j, for all $i < j \leq N$) are obtained as the output from the solver.

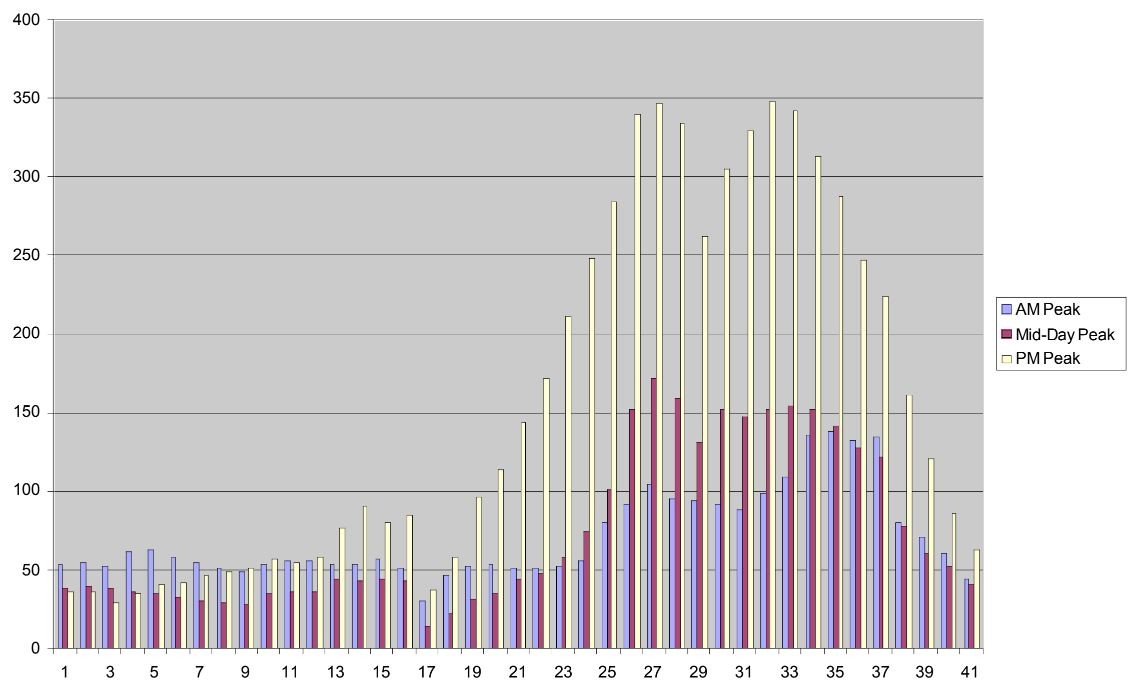
6.3 Performance Estimation and Comparison

The requirements for the software tool is given in Appendix B. The C program implementing the requirements is listed in Appendix C.4.

7. NUMERICAL RESULTS FOR THE EXAMPLE CORRIDOR

Based on the data of Appendix A.2 and the constrained entropy maximization method, we obtained the maximum-entropy (or the "flattest" distribution) of OD trips that satisfies the on and off passengers counts given in Appendix A.2. In turn, we obtained estimates of section demand for all 41 sections and for all three peak periods; they are summarized in Figure 1.

Figure 1. Hourly Demand by Section



The result of evaluation and comparison regarding the first three criteria for each of the four route structures tested can be summarized in Tables 2, 3 4, 5 and 6.

Table 2: Three Major Performance Measures for 2 Partial Routes

Performance Measures	Light Rail	Shuttle-centered Busway	Shuttle-centered ABUS
Total Equipment Time	2040	1002	1002
Total Labor Time	510	1002	510
Total Fuel Cost	N/A	3006	2858

Table 3: Three Major Performance Measures for 2 Southern Partial Routes

Performance Measures	Light Rail	Shuttle-centered Busway	Shuttle-centered ABUS
Total Equipment Time	2040	974	974
Total Labor Time	510	974	510
Total Fuel Cost	N/A	2922	2783

Table 4: Three Major Performance Measures for 1 Long Partial Route

Performance Measures	Light Rail	Shuttle-centered Busway	Shuttle-centered ABUS
Total Equipment Time	2040	828	828
Total Labor Time	510	828	510
Total Fuel Cost	N/A	2484	2389

Table 5: Three Major Performance Measures for 1 Short Partial Route*

Performance Measures	Light Rail	Shuttle-centered Busway	Shuttle-centered ABUS
Total Equipment Time	2040	684	684
Total Labor Time	510	684	510
Total Fuel Cost	N/A	2052	2000

* The capacity of this configuration is slightly less than the demand of PM Peak.

Table 6: Three Major Performance Measures for 1 Partial Route in South Half

Performance Measures	Light Rail	Shuttle-centered Busway	Shuttle-centered ABUS
Total Equipment Time	2040	792	792
Total Labor Time	510	792	510
Total Fuel Cost	N/A	2376	2291

As mentioned earlier, we inflate the current demand to study how much more demand can the individual route structures can accommodate. The highest amount of demand a route structure can accommodate depends on the time period. We summarize the results in Table 7.

Table 7: The Highest Demand That Can Be Satisfied by the Four Route Structures (In increment of 25% of the Current Demand)

Route Structure	AM Peak	Mid-day Peak	PM Peak
Two Central Partial Routes	225%	225%	125%
Two Southern Partial Routes	400%	475%	250%
One Partial Route (long)	225%	225%	125%
One Partial Route (short)	200%	175%	Not sufficient for even the current demand
One Partial Route – Southern Half	400%	325%	150%

Based on these numerical results, it is clear that the shuttle-center ABUS system can be drastically more efficient than the light-rail system in terms of equipment requirement. For example, to satisfy the current demand, only 792 vehicle-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 2040 vehicle-minutes required by the corresponding light-rail operations. This translates into over 60% reduction of equipment requirement.

It is also clear that the shuttle-centered ABUS system is drastically more efficient than the shuttle-centered (conventional) busway. For example, to satisfy the current demand, only 510 driver-minutes are required per hour of shuttle-centered ABUS operations with one shuttle and the southern partial route vs. the 792 driver-minutes required by the corresponding conventional operations. This translates into over 36% reduction of labor requirement.

It is also clear that routing is an important factor. For example, shuttle-centered ABUS or busway with the shuttle and the short partial route would not suffice even for the current demand. This is because the short partial route does address the relatively high demand in the southern part of the system. If two partial routes are required for non-efficiency reasons, e.g., to minimize the need for passenger intra-convoy transfer, then the two partial routes should definitely be located where demand is high. The route structure of Two Southern Nested Partial Routes would be better. In particular, this structure can accommodate not only the current demand but also the 250% of the current demand while the route structure of Two Central Nested Partial Routes can accommodate up to only 125% of the current demand.

While the shuttle-center ABUS offers drastic improvement over the light-rail and shuttle-centered busway operations, its fuel advantage over the shuttle-centered busway ranges from between 2.5% to 5% and hence is not significant.

8. CONCLUDING REMARKS

It is clear that the new operating concept has the potential of offering drastic improvement in operational efficiency than its light-rail and conventional counterparts on the mainline operations alone, not to mention on integrated mainline and local operations. Further evaluations and comparisons of these system concepts, with or without integration with the local operations, are worthy future research topics. System simulation may be required. The design of a fail-safe automated driverless closely spaced bus following is a worthy subject for future research. Although routing and scheduling for either bus systems can be optimized, such important tasks require consideration of many factors that are not easy to quantify. Due to the required user-friendliness for any bus system, bus operations must be kept simple from the rider's perspective. The proposed concept is simple by design; the operating scenarios developed and selected for the comparisons are also simple. The simplicity also facilitates comparison of the proposed operations with the corresponding systems.

REFERENCES

Aso, Makoto; Suzuki, Toshihiko, "Automated steering control for the Intelligent Multimode Transit System," IEEE Intelligent Vehicles Symposium (2000 : Dearborn, Mich.). IV 2000 : proceedings of the IEEE Intelligent Vehicles Symposium 2000, p. 590-595.

Bishop, Richard, "Automatic for the people : computer-driven transit buses just around the corner," ITS world. Vol. 5, no. 2, pp. 16-17, March/April 2000.

Chen, C.-Y. and Litkouhi, B., "Technological Challenges in the Development of AHS," ITS Journal, Vol. 4, No. 1-2, pp. 81-100, 1998.

DeMarco, V. R., "Dual-Mode System Development Program of the Urban Mass Transportation Administration," Proceedings of International Conference on Dual Mode Transportation, Washington, D.C., 1974; Transportation Research Board Special Report No. 170, **:** Transportation Research Board, National Research Council, Washington, D.C..

Fang, S.-C. and Tsao, H.-S.J., (1995) "Linearly-Constrained Entropy Maximization Problem with Quadratic Costs and Its Applications to Transportation Planning Problems," Transportation Science, Vol. 29, No.4, pp. 353-365, 1995.

Fang, S.-C., Tsao, H.-S.J., and Rajasekera, J.R., (1997) Entropy Optimization And Mathematical Programming, Kluwer Academic Publishers, September, 1997.

Shladover, S.E., (1995), "Review of State of Development of Advanced Vehicle Control Systems (AVCS)," International Journal of Vehicle Mechanics and Mobility, Vol. 24, pp. 551-595, 1995.

Shladover, S.E., (2000), “Progresive Deployment Steps Leading Toward an Automated Highway System,” Transportation Research Record 1727, pp. 154-161, Transportation Research Board, 2000.

Shladover, S. E, “PATH demonstrates automated bus rapid transit technologies,” Intellimotion. Vol. 10, no. 3, Univ. of Calif., Berkeley, Inst Transp Studies, 2003.

Tan, Han-Shue (2003), “Develop precision docking function for bus operation,” PATH research report ; UCB-ITS-PRR-2003-11, California PATH Program, Institute of Transportation Studies, University of California, Berkeley CA.

Tsao, H.-S. J. (1995a), "Stage Definition for AHS Deployment and an AHS Evolutionary Scenario", Intelligent Vehicle Highway Systems Journal, Vol.2(4), pp. 359-382, 1995.

Tsao, H.-S. J. (1995b), "A Staggered-Diamond Design for Automated/Manual-HOV Highway-to-Highway Interchanges and Constraints on AHS Design for Accommodating Automated Highway Change", Intelligent Vehicle Highway Systems Journal, Vol.2(3), pp. 281-292, 1995.

Tsao, H.-S. J. (1995c), "Constraints on Initial AHS Deployment and the Concept Definition of a Shuttle Service for AHS Debut", IVHS Journal, Vol. 2(2), 159-173, 1995.

Tsao, H.-S. J. (1998a), "From the Guest Editor," Special Issue of the ITS Journal on Automated Highway Systems, Vol. 4, Nos:1-2, 1998.

Tsao, H.-S.J. (1998b), "An Axiomatic Approach to Developing Partial Automation Concepts for Deployment of Automated Highway Systems and Partial Invocation of Vision-Based Lane-Keeping and Adaptive Cruise Control," *Transportation Research Record*, No. 1651, 1998.

Tsao, H.-S. J. (2001), " A Framework for Evaluating Deployment Strategies for Intelligent Transportation Systems", Intelligent Transportation Systems Journal (ITS Journal), Vol. 6, pp. 141-173, 2001.

Tsao, H.-S.J. and Botha, J.L. (2001), “Operating Concepts for Urban Bus Automation and Inter-city Truck Automation,” the Proceedings of the 2001 IEEE ITS Conference, Oakland, California.

Tsao, H.-S. J., Botha, J.L. (2003), “Definition and Evaluation of Bus and Truck Automation Operations Concepts: Final Report,” California PATH Research Report UCB-ITS-PRR-2003-19, May 2003.

Tsao, H.-S. J. and Hall, R.W., "A Probabilistic Model for AVCS Longitudinal Collision/Safety Analysis", Intelligent Vehicle Highway Systems Journal, Vol.1, No. 3, pp. 261-274, 1994.

Tsao, H.-S.J., Hall, R.W., Shladover, S.E., Plocher, T.A. and Levitan, L.J., "Human Factors Design of Automated Highway Systems: First Generation Scenarios," US-DOT FHWA Report FHWA-RD-93-123 (1993).

Tsao, H.-S. J.,, T.A. Plocher, W.B. Zhang, and S.E. Shladover, "Human Factors Design for Automated Highway Systems: Second Generation Scenarios", draft US-DOT FHWA Report, under FHWA Contract No. DTFH61-91-C-00100, (Jan. 14, 1994).

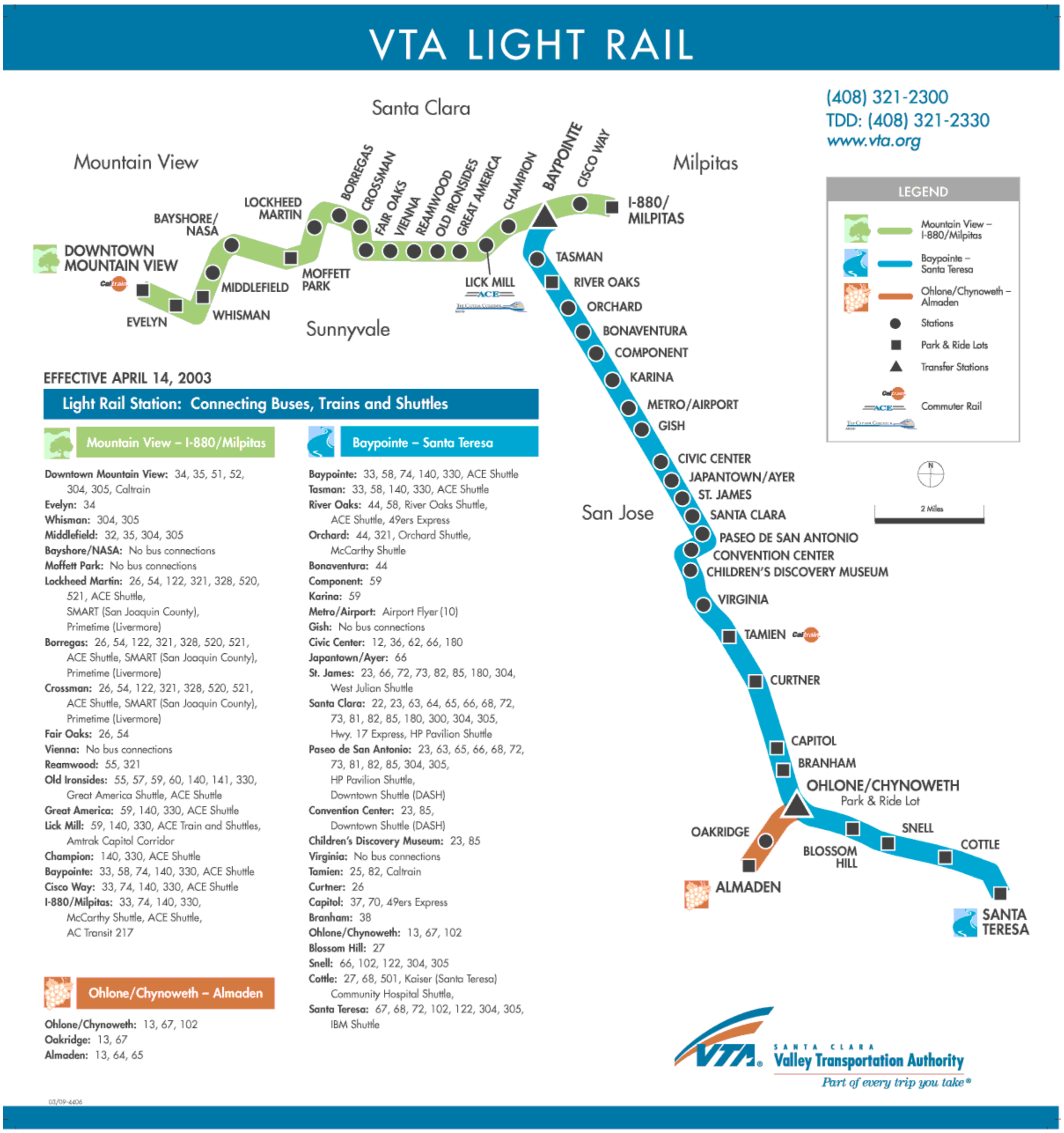
Tsao, H.-S. J. and Ran, B., "Driver Intelligence Replacement in a Decision-Oriented Deployment Framework for Driving Automation", Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida (1996).

Ulmer, Berthold, "CHAUFFEUR: From Concept to Realization," 3rd International Workshop on Vehicle-Highway Automation, Toronto, Canada, Nov. 11-12, 1999.

Appendix A: The Reference Light-rail System – Santa Clara County Light Rail System Operated by the Valley Transportation Authority of Santa Clara Country, California

Appendix A.1: Light-rail System Map

We treat the commute corridor with 42 stations from Downtown Mountain view to Santa Teresa as one line, rather than the actual two lines connected at the Baypointe Station.



Appendix A.2: Passenger Volumes

Light Rail Data: Passenger arrival/departure data for the AM, PM, Mid-day and Off-peak time. NOTE: These are total counts for these periods, instead of hourly rates.

Station Number	Corridor Stations	AM PEAK 5:30 to 8:30		MID PEAK 8:30 to 14:30		PM PEAK 14:30 to 17:30		OFF PEAK 17:30 to 5:30	
		ON	*OFF*	*ON*	*OFF*	*ON*	*OFF*	*ON*	*OFF*
1	*MOUNTAIN VIEW*	159	0	228	0	113	0	111	0
2	*EVELYN*	7	0	11	0	4	0	1	3
3	*WHISMAN*	12	10	10	15	7	22	6	25
4	*MIDDLEFIELD*	47	17	19	24	22	5	22	4
5	*BAYSHORE NASA*	6	0	5	4	21	1	7	2
6	*MOFFETT PARK*	0	12	2	11	8	1	6	1
7	*LOCKHEED MARTIN*	9	18	21	23	29	16	11	10
8	*BORREGAS*	2	10	3	11	13	6	19	0
9	*CROSSMAN*	0	7	5	15	13	1	10	0
10	*FAIR OAKS*	30	8	64	8	37	14	33	16
11	*VIENNA*	13	5	23	5	7	12	12	16
12	*REAMWOOD*	9	6	15	6	15	0	10	2
13	*OLD IRONSIDES*	20	23	89	34	75	19	54	13
14	*GREAT AMERICA*	2	1	20	20	56	9	42	5
15	*LICK MILL*	23	11	34	19	17	41	18	27
16	*CHAMPION*	4	23	13	18	16	3	6	1
17	*BAYPOINTE*	54	0	39	0	57	0	28	0
18	*TASMAN*	46	1	46	1	61	0	44	1
19	*RIVER OAKS*	30	9	57	13	112	5	34	1
20	*ORCHARD*	12	7	30	7	48	0	25	1
21	*BONAVENTURA*	14	24	93	21	90	8	43	4
22	*COMPONENT*	10	12	34	9	76	2	24	1
23	*KARINA COURT*	35	30	81	25	122	11	73	14
24	*METROAIRPORT*	20	12	111	32	127	26	79	24
25	*GISH*	69	6	171	33	132	30	69	27
26	*CIVIC CENTER*	81	52	404	132	259	119	95	62
27	*JAPANTOWNAYER*	43	10	138	37	54	35	39	34
28	*ST JAMES*	20	46	106	170	72	114	27	53
29	*SANTA CLARA*	126	124	263	476	198	432	169	257
30	*SAN ANTONIO*	52	77	321	188	243	123	138	70
31	*CONVENTION CENTER*	14	37	162	213	163	89	108	38
32	*DISCOVERY MUSEUM*	30	8	51	24	65	21	40	21
33	*VIRGINIA*	37	11	36	35	23	43	19	28
34	*TAMIEN*	113	35	127	137	104	203	90	90
35	*CURTNER*	50	46	61	128	49	126	25	97
36	*CAPITOL*	55	77	74	171	60	185	38	110
37	*BRANHAM*	18	20	13	63	9	89	7	54
38	*OHLONE-CHYNOWETH*	13	175	52	312	40	249	22	135
39	*BLOSSOM HILL*	8	41	16	118	8	133	5	76

40	*SNELL*	3	32	18	81	7	103	8	66
41	*COTTLE*	3	54	2	84	1	67	2	61
42	*SANTA TERESA*	0	122	0	238	0	204	0	196

Appendix A.3

Approximate Distances Bbetween Two Neighboring Stations.

Station Number	Distance *(in miles)*	**Station Number**	Distance *(in miles)*
1-2	*0.6*	22-23	*0.6*
2-3	*0.5*	23-24	*0.4*
3-4	*0.7*	24-25	*0.6*
4-5	*1.5*	25-26	*0.8*
5-6	*0.3*	26-27	*0.5*
6-7	*0.7*	27-28	*0.5*
7-8	*0.4*	28-29	*0.2*
8-9	*0.5*	29-30	*0.2*
9-10	*0.6*	30-31	*0.5*
10-11	*0.6*	31-32	*0.3*
11-12	*0.5*	32-33	*0.6*
12-13	*0.3*	33-34	*0.8*
13-14	*0.7*	34-35	*1.3*
14-15	*0.7*	35-36	*1.5*
15-16	*0.8*	36-37	*0.6*
16-17	*0.6*	37-38	*0.7*
17-18	*0.3*	38-39	*0.7*
18-19	*0.5*	39-40	*0.8*
19-20	*0.6*	40-41	*0.8*
20-21	*0.5*	41-42	*0.8*
21-22	0.4		

APPENDIX B: FUNCTIONAL REQUIREMENTS OF THE ALGORITHMS FOR EVALUATING AND COMPARING THE OPERATIONAL EFFICIENCY OF THE THREE SYSTEMS

OBJECTIVES OF THE BUS COMPONENT OF THE RESEARCH PROJECT:

Alternatives:

- automated bus operations, on current light-rail right of way or on planned new light-rail lines

to the following two conventional alternatives:

- conventional light-rail system (involving downtown segments)
- Non-automated exclusive busways.

Benefit and cost categories:

Infrastructure Costs:
- Mainline

Operator Costs:
- Equipment cost: Capital and Maintenance
- Operating cost: Labor, Fuel

User Cost:
- Passenger Travel Time

The deployment site serving as a reality check:

To make the ABUS operating concepts as clear as possible, we will use the commute corridor served by the Santa Clara County Light-rail System operated by the Valley Transportation Authority (VTA) as the hypothetical testing ground.

SCOPE OF THE METHODOLOGY:

Same as the Project Scope

Benefit and cost categories:

Operator Costs:
- Equipment cost: Capital and Maintenance
- Operating cost: Labor, Fuel
- Passenger travel time

Note: The total costs of these categories for the three different alternatives will be estimated by multiplying unit costs by the corresponding total amounts. The units will be addressed below.

Note: The following is beyond the scope of this methodology:

Infrastructure Costs:
- Mainline

THE RELATIONSHIP BETWEEN THIS SCOPE AND THE REST OF THE BUS COMPONENT OF THIS PROJECT: SEPARATE AND PARALLEL, BUT COODINATED THROUGH THE COMMUTE CORRIDOR

In essence, the evaluation and comparison is decoupled into two separate and parallel but coordinated activities. The primary link between the two is the commute corridor. Particular characteristics of importance include:

- The right-of-way of the Santa Clara Light-rail System
- From Mountain View to Santa Teresa, but as a single corridor. (The system consists primarily of two lines, although a third line connects Ohlone/Chynoweth to Almaden, with only one station in between. One of the two lines connects Mountain View to the Great Mall; the Tasman Station near the intersection of the North First Avenue and the Tasman Driver is a station in between. But that line has not been completed yet, and the current service on that line stops a little beyond the Tasman Station. Another line connects the Tasman Station and the Santa Teresa station. Although transfer is needed between these two lines at the Tasman Station, we consider them as one corridor from Mountain View to Santa Teresa. This is because the stations served by the current system virtually form one corridor. Also, since we deal with alternatives comparable to an unprotected light-rail system and the portion of the Santa Clara Light-rail system south of Children's Museum is built within a protected wall or fence, we consider the corridor along the light-rail system between Mountain View and Children's Museum. Note that the primary differences between the protected and unprotected systems include the infrastructure cost and safety.)

THE METHODOLOGY: THE INPUT

Current corridor data:
- For every segment,
    - Length
    - Presence of a station or not
    - Volume data: Incomplete data sets
        - Origin data for some stations
        - Origin data for some cities (i.e., group of stations)
        - Destination data for some stations
        - Destination data for some cities (i.e., group of stations)
        - Origin-destination data for some station pairs

- Origin-destination data for some city pairs
- Peak vs off-peak data for some origins

Demand Inflation Factors: Current demand will be inflated by increments of 25% of the current demand, and all the analyses performed for the current demand will be repeated for these inflated demand levels.

Temporal Demand Split Factor: Peak vs. off peak for volume data not already split by the data providers (Hours and volume splits are to be determined by Botha or jointly and their implementation should be straightforward.)

Unit-cost data:

- Vehicle Travel Time or, Equivalently, Equipment Cost (light-rail car, regular bus or ABUS): in vehicle-revenue-hour, where vehicle could be a light-rail car, a regular bus or an ABUS), i.e., in
  - car-revenue-hour,
  - bus-revenue-hour and
  - ABUS-revenue-hour.
- Passenger Travel Time: per passenger-hour
- Labor (Driver): per vehicle-driver-revenue-hour
  - Note: We do not distinguish the labor costs among light-rail, ABUS and conventional-bus driving. If the driver cost associated with ABUS is significantly different from its light-rail and conventional bus counterparts, then we need to track
    - Labor (ABUS Driver): per ABUS-driver-revenue-hour
    - Labor (Light-rail Driver): per light-rail-driver-revenue-hour
    - Labor (Conventional Driver): per conventional-bus-driver-revenue-hour
- Fuel: per vehicle-revenue-mile at the average design speed (40 mph as the default design speed for all three systems or also at other speeds.
  - Three different parameters:
    - Per light-rail-car-revenue-mile
    - Per ABUS-revenue-mile
    - Per conventional-bus-revenue-mile
  - Note: The reduction in fuel consumption due to closely-spaced convoying of automated trucks will be expressed in terms of a percentage of the consumption without it.

THE METHODOLOGY: ASSUMPTIONS

Assumptions:

- The local portions of the benefits and costs associated with the three alternatives are assumed to be equal, and they are cancelled in the comparisons.
- It suffices to focus on the mainline.

Several design options selected are to be discussed later.

THE METHODOLOGY: AN OPTIONAL INTERMEDIATE STEP

Origin and Destination trip estimation based on the volume data obtained so far from various sources:

Scope: Origins and destinations refer to the start and end stations of the trips on the mainline, i.e., the origin and destination stations. They do not refer to the locations of the riders' true trip origins beyond the mainline and their true destinations beyond the mainline.

- Approach: maximum-entropy approach

This methodology works with any given set of OD trip data. It is nice to use the volume data collected so far to estimate OD trip numbers along the corridor under consideration. Since we are not solving the problem for the selected site, but are using the real site for guidance, especially for reality check at the concept level, this step is actually optional. We can assume a reasonable OD trip pattern, or can run the program for multiple OD trip patterns.

THE METHODOLOGY: THE OUTPUT

Given a set of OD trip volume data along the corridor (mainline only), a set of computer program will be developed in C or C++ to provide, for each of the three options

- automated bus operations, on current light-rail right of way or on planned new light-rail lines
- conventional light-rail system (involving downtown segments)
- Non-automated exclusive busways,

the following numerical values:

Operating cost category: Applicable for all three alternatives, with "vehicle" to be replaced by light-rail-car, ABUS and conventional-bus, respectively.

- Total Vehicle Travel time
  - Measure: total number of vehicle-revenue-hours - <u>one number</u>
- Total Passenger Travel Time
  - Measure: total number of passenger-hours – <u>one number</u>
- Total Labor Requirement
  - Measure: the total number of driver-revenue-hours - <u>one number</u> (This is different from vehicle-revenue-hours for the Busway option because one driver is required for every bus.)
- Total Fuel Requirement:

    - Measure: the total fuel cost = total vehicle-revenue-miles x unit cost for the vehicle-fuel - <u>one number</u> (Operations at a constant design speed throughout the corridor mainline assumed)
- Equipment cost: Capital and Maintenance
    - Measure: the total number of vehicle-revenue-hours - <u>same as travel time measure</u>

Four Numbers: Four Performance Values (“Vehicle” = light-rail car, ABUS or conventional bus)

- Total Vehicle Travel Time (i.e., Total Equipment Cost): vehicle-revenue-hour
- Total Passenger Travel Time: passenger-hour
- Total Driver Labor Requirement: driver-revenue-hour
- Total Fuel Requirement: vehicle-revenue-miles (constant speed through the corridor mainline assumed)

These four are recurring daily costs.

THE METHODOLOGY: A HIGH-LEVEL INTRODUCTION TO THE APPROACH

The problem is too big to be solved as one optimization problem. We use a simpler approach for easier comparisons. A C Program will be developed to perform the required analysis.

Light-rail Operations

Take the current schedule of the Santa Clara Light-rail System as it is and figure out the four performance values.

ABUS Operations

<u>Approach</u>

- Discrete sets of routes and the companion schedules as discrete choices: The design options to be discussed later include one about routes and schedules, particularly about “partial-routes” and the companions schedules. Many possible sets of such partial routes and the companion schedule exist. We will consider a small number of possible sets. These sets may be characterized by a small number of parameters.

- Estimate the operating cost resulting from the satisfaction of the OD demand for <u>one day</u>. (To get the operating cost for one year, for example, just multiply these values by 365.)

- Selection among the sets of partial routes and the companion schedules the best one as the base for ABUS benefit-cost calculation.

There exist a number of different ways to operate an ABUS system. We will briefly state design options selected for this methodology later.

Busway Operations

We operate the conventional buses on the dedicated busway as close to the way we operate the ABUS as possible. Note that the major differences center around closely-spaced automated convoying, particularly

- One driver per bus: a major difference in labor cost
- No fuel savings due to reduction in air resistance: a major difference in fuel cost
- No automated lateral control: a major difference in the requirement for the width of right-of-way
- Difficulty in coordinating bus traffic on the busway with the bus-priority signaling.

The last two effects will be addressed separated. The absence of automated lateral control pertains to right-of-way sufficiency and cost; the disturbance to surrounding traffic due to bus priority signal is addressed in a separate document.

THE METHODOLOGY: ABUS SYSTEM DESIGN OPTIONS SELECTED

We add features one at a time for conceptual clarity and for ease of comparison.

Mimicking light-rail

Mimic the light-rail system operations with closely-spaced automated bus convoys (abbreviated as CABCs), with each CABC mimicking a light-rail train of multiple light-rail cars and operating exclusively on the mainline right-of-way. The major difference between ABUS and light-rail operations result from

- difference in equipment
- difference in fuel.

Although the difference lies in the difference in the unit costs, we still have to figure out the common total vehicle-revenue-hours.

One Shuttle Route Between the Two ends, with One Automated Bus Supervised by a Driver

To capitalize the flexibility of bus operations, operate an automated mainline shuttle bus (containing only one automated bus) between the two ends of the mainline at the same light-rail schedule.

Partial Routes between Two Stations, Intended to Serve Local and Mainline Needs on the Same Bus

There may be other “partial bus routes” connecting two stations on the mainline. All the buses serving those partial bus routes may also be serving local collection and distribution, and the routing and scheduling of the buses serving those routes will be such that the same bus can be used to collect passengers on the city streets near the origin station, travel along the mainline sections involved and then distribute passengers near the other station. However, these buses will be and can only be traveling with an automated mainline shuttle bus in a convoy. Such a shuttle bus plus the companion automated buses form a closely-spaced automated bus convoy.

A bus serving a partial route may enter a station first and wait for such a convoy or may join such a convoy at a station from behind. There will be one driver per convoy, and he or she will be the driver driving the shuttle of the convoy out of one end of the corridor mainline. These mainline shuttle drivers are dedicated to mainline driving; drivers of the partial routes perform the driving on city streets only and not on the mainline. After a driver drives a bus equipped with automation into a station and after the bus is electronically linked with a convoy, he or she leaves the bus and prepares to drive a bus arriving with a convoy but leaving the mainline to enter city streets. When a convoy is reformed and is ready to leave the station, the driver moves to the first bus of the convoy if he or she is not already on the first bus of the convoy.

The routing and scheduling of the partial routes will be a source of possible benefit over the light-rail system because such partial routes have the potential of providing the service where it is needed. (With a light-rail system, typical operations include a light-rail train with a fixed number of cars traveling back and forth along the corridor, and they may have many empty seats particularly when traveling sections near the two ends.)

The possible increase of ridership is beyond of the scope of this study. First of all, there are no data about the origins and destinations of those trips part of which involves travel on the current light-rail system. Second, ridership estimation requires the development of demand models, which is clearly a very complex issue along. The possible benefit will be pointed out, but will not be quantified. Therefore, the true benefit of ABUS operations may be higher than our estimate.

Perhaps more importantly, we do not attempt to develop an optimal operating plan for ABUS operations (or busway operations) and then compare the optimal performance with the performance of the other two alternatives. Optimization for either the AHS operations or the busway operations is a complex issue by itself. Also, this optimization requires data about commuters’ true trip origins and destinations.

Intra-convoy Transfer:

Once on the mainline, a passenger who is on a partial-route bus but wants to travel to a station beyond what the bus will reach can transfer to, for example, the end-to-end shuttle. Note that the transfer incurs no wait at all because all the passenger needs to do is to alight the partial-route bus and board the end-to-end shuttle of the same convoy. In fact, if the passenger's trip destination is off the mainline, there is another partial route serving the destination and another bus of the convoy is serving that partial route, then the passenger can transfer to that bus, again by capitalizing on the intra-convoy transfer and without incurring any wait.

Section-dependent Capacity to Satisfy Section-dependent Demand: A Major Source of Improvement in Operational Efficiency (over the Light-rail) Enabled by Electronic Linkage of Buses and Intra-convoy Transfer

This flexibility actually plays a pivotal role in the ability of ABUS system to offer higher operating efficiency than the light-rail system. A major operating inefficiency of light-rail is that it typically runs a train from one end to the other with the same number of light-rail cars regardless of the variability in demand along the mainline. In other words, the uniform capacity of the light-rail throughout the whole mainline is to meet the demand of the heaviest travel section. However, due to the flexibility of electronically linked automated buses and the flexibility of intra-convoy transfer, it is much easier for the ABUS system to provide just enough section-dependent capacity to meet the section-dependent demand.

Although intra-convoy transfer incurs no wait time, it may cause some inconvenience to the rider. Therefore, it would be good not to require more than a couple of such transfers. One way to achieve this is to limit the number of partial routes. In designing partial routes, we will limit the number of such partial routes to three or four.

Again, we do not explicitly study the quantitative impact of this flexibility on the ridership.

One example set of partial routes and the companion schedule is as follows.

Consider the Santa Clara Light Rail system between Mountain View and the Children's Museum south of the downtown San Jose area, and treat the corridor as one continuous mainline, as discussed earlier. The routes are

- End-to-end route from Mountain View to Children's Museum, at the same headway as the current light-rail system
- Three partial routes, all at the same headway of the end-to-end route:
    - From Children's Museum north up to 2/3 of the way to Mountain View
    - From Children's Museum north up to 1/3 of the way to Mountain View
    - From the 1/4 point north of the Children's Museum to 3/4 point of the Children's Museum.

Although these partial routes may not be mathematically optimal in terms of allocating optimally capacity to where it is needed, such routes should be simple and easy to remember for the user.

Like the current light-rail schedule, the headway may vary between peak and non-peak hours. Note that there may still be local bus services serving a local area and serving to connect the local travelers to the mainline ABUS system.

THE METHODOLOGY: A MORE DETAILED DESCRIPTION

We first discuss the algorithms needed to estimate the performance and cost of the ABUS alternative and then point out the difference between the ABUS alternative and the busway alternative as well as the difference between the ABUS alternative and the light-rail alternative.

Algorithms Estimating the Performance Values: The ABUS Alternative

A small number of sets of partial routes and companion schedules will be chosen and the resulting performance values estimated. For any such given set, a weighted sum of the four totals for the four performance values will be calculated and will be used as the overall total operating cost associated with the corresponding set of partial routes and the companion schedule, where the weights are their unit costs. The best one, i.e., the lowest-cost one and the corresponding set will be selected as the performance values for this alternative.

Step 0: For every set of partial routes and the companion schedule, perform the following steps:

<u>What follows will be repeated for every set of partial routes and the companion schedule. But, the actual sets will be determined dynamically as we improve upon the one stated earlier.</u>

We now describe how to obtain the four performance values based on a given set of partial routes and the companion schedule.

Step 1: Estimate the Total ABUS Travel Time (i.e., Total Equipment Cost): vehicle-revenue-hour (i.e., ABUS-revenue-hour)

This Total ABUS Travel Time can be obtained by multiplying, for each route, the travel time of the route by the total number of trips made for that route in one day and summing the total over all the routes. We assume that the travel time for each section is deterministic and known and is equal to the distance of that section divided by the design (average) speed.

Step 2: Estimate Total Passenger Travel Time: passenger-hour

First calculate the average travel time for every possible OD. It consists of a wait time at the mainline origin that is equal to half of the headway and the time required to travel on the mainline to the mainline destination.

Based on the estimated OD trip numbers along the mainline, calculate the weighted sum of the average travel time weighted by the OD trip numbers.

Step 3: Calculate the Total Driver Labor Requirement: driver-revenue-hour

The Total Driver Labor is simply the total number of the end-to-end shuttle trips because there will be exactly one driver per convoy and the number of convoys in one day is simply the number of the end-to-end shuttle trips.

Step 4: Total Fuel Requirement: vehicle-revenue-miles (constant speed through the corridor mainline assumed)

Assume that automated buses travel at a constant design speed or simply deal with the average speed of a bus.

Under this assumption, the fuel consumption of an automated bus depends on the distance travel and whether it is part of a convoy and, if so, where it is with respect to the other buses in the same convoy. However, the fuel consumption of a convoy depends only on its size.

First focus on a particular end-to-end shuttle bus traveling from the Children's Museum to Mountain View first. As the automated bus travels, other automated buses may join it to form a convoy or may split from the convoy and leave the mainline. Note that all buses move in such convoys and any such convoy contains an end-to-end shuttle bus.

Now, calculate the evolution of the convoy size as a given convoy travels from one end to the other based on the given partial routes and the companion schedule.

We calculate the fuel requirement by a given convoy traveling through a given section of given length. Summing over all these fuel requirement values over all sections and all convoys produces the total amount of fuel consumption.

Step 5: Calculate the number of passengers aboard each of the convoy in each of the sections, and check if the number exceeds the capacity of the convoy.

If the number of passengers exceeds the capacity of the convoy (calculated in Step 4), then the set of partial route and the companion schedule is not acceptable and should be discarded or improved.

Step 6: Calculate the overall performance of the set of partial routes and the companion schedule under consideration by weighting the four performance values weighted by their unit costs.

Step 7: Select the best among the sets of partial routes and the companion schedule studied.

This group of algorithms produces the following four numbers

- Total ABUS Travel Time (i.e., Total Equipment Cost): ABUS-revenue-hour
- Total Passenger Travel Time: passenger-hour
- Total ABUS Driver Labor Requirement: ABUS-driver-revenue-hour
- Total ABUS Fuel Requirement: ABUS-revenue-miles (constant speed through the corridor mainline assumed)

and a weighted sum of these values weighted by their unit-costs.

<u>The Final Output Template for the ABUS Alternative:</u>

	Current Demand	150% Inflation	200% Inflation	Unit Cost
Total ABUS Travel Time (i.e., Total Equipment Cost): ABUS-revenue-hour				
Total Passenger Travel Time: passenger-hour				
Total ABUS Driver Labor Requirement: ABUS-driver-revenue-hour				
Total ABUS Fuel Requirement: ABUS-revenue-miles				

<u>Templates for Key Intermediate Results:</u>

Basic ABUS Link Information (One Direction; Daily)

ABUS Section #		
Length of Section		
Speed of Section – Design Speed: Miles per Hour		

Basic ABUS Access/Egress Points

ABUS Access Point #		
Location		

ABUS Routes and Schedules: (One Direction)

Route #		
Route Origin Access Point		
Route Destination Access Point		
Headway		

Note: Keep route structure and schedule separate from other parts of the program, at least as much as possible, because new routes and schedules may be studied as we gain experience on the relationship of performance of the AHS as a function of the routes and schedules. Also, keep the route structure and the schedule separate from each other if possible.

Algorithms Estimating the Performance Values: Busway

Operate the system the same way as the ABUS system to the maximum possible extent. We still organize the conventional buses into convoys, but not closely-spaced convoys. Perhaps more importantly, driverless operations cannot be done. Therefore, the primary differences between the ABUS and Busway operations are the labor cost and fuel cost. Also, because the buses cannot form closely-spaced convoy, they require more space when traveling and more time to allow intra-convoy transfers. Moreover, when combined with bus priority signaling, they may disturb the surrounding traffic more than their ABUS counterparts.

All the steps are identical to those of the algorithms for the ABUS alternative, except for the following:

Step 3: Calculate the Total Driver Labor Requirement: driver-revenue-hour

For each scheduled bus, calculate the travel time from the mainline origin to the main destination. Sum up all the travel time over all scheduled buses.

Step 4: Total Fuel Requirement: vehicle-revenue-miles (constant speed through the corridor mainline assumed)

For each scheduled bus, calculate the travel distance. Sum up these travel distances over all scheduled buses.

This group of algorithms produces the following four numbers

- Total Conventional Bus Travel Time (i.e., Total Equipment Cost): conventional-bus-revenue-hour
- Total Passenger Travel Time: passenger-hour

- Total Conventional Bus Driver Labor Requirement: conventional-driver-revenue-hour
- Total Fuel Requirement: conventional-bus-revenue-miles (constant speed through the corridor mainline assumed)

and a weighted sum of these values with their unit-costs as the weights.

	Current Demand	150% Inflation	200% Inflation	Unit Cost
Total Conventional Bus Travel Time (i.e., Total Equipment Cost): conventional-bus-revenue-hour				
Total Passenger Travel Time: passenger-hour				
Total Conventional Bus Driver Labor Requirement: conventional-driver-revenue-hour				
Total Fuel Requirement: conventional-bus-revenue-miles				

Algorithms Estimating the Performance Values: The Light-rail Alternative

We only point out the algorithmic differences between the ABUS alternative and the light-rail alternative.

Step 1: Estimate the Total Light-rail Travel Time (i.e., Total Equipment Cost): vehicle-revenue-hour (i.e., light-rail-car-revenue-hour)

This Total Light-rail Travel Time can be obtained by multiplying the total number of light-rail cars traveling between the two ends in one day by the travel time between the two ends.

Step 4: Total Fuel Requirement: vehicle-revenue-miles (constant speed through the corridor mainline assumed)

Assume that light-rail trains travel at a constant design speed or simply deal with the average speed of a light-rail train.

Under this assumption, the fuel consumption of an automated bus depends on the distance travel and whether it is part of a train and, if so, where it is with respect to the other light-rail cars in the same train. However, the fuel requirement of a train depends only on its size.

Multiply the fuel requirement of a train of a particular size by the number of such trains scheduled per day, and then sum over all possible sizes. Multiply the result by the distance of the mainline between the two ends.

Step 6: Calculate the overall performance of the current light-rail system by weighting the four performance values with their unit costs as the weights.

This group of algorithms produces the following four numbers

- Total Light-rail Travel Time (i.e., Total Equipment Cost): Light-rail-care-revenue-hour
- Total Passenger Travel Time: passenger-hour
- Total Light-rail Driver Labor Requirement: light-rail-driver-revenue-hour
- Total Light-rail Fuel Requirement: light-rail-revenue-miles (constant speed through the corridor mainline assumed)

and a weighted sum of these values with their unit-costs as the weights.

	Current Demand	150% Inflation	200% Inflation	Unit Cost
Total Light-rail Travel Time (i.e., Total Equipment Cost): Light-rail-care-revenue-hour				
Total Passenger Travel Time: passenger-hour				
Total Light-rail Driver Labor Requirement: light-rail-driver-revenue-hour				
Total Light-rail Fuel Requirement: light-rail-revenue-miles				

APPENDIX C: COMPUTER PROGRAMS

Appendix C.1: Computer Programs for Soliciting User Input about Mainline Configuration and Partial Routes

```
#include <stdio.h>
#define NUM_DEFAULT_PR    3

main()
{
    int pr = 0;
    int *org, *dst;
    int i;
    char choice, buffer[100];
    int stations = 0;
    int speed = 1;
    int ampeak = 10;
    int pmpeak = 15;
    int midday = 15;
    int offpeak = 20;

    printf("Enter the number of stations: ");
    gets(buffer);
    sscanf(buffer, "%d", &stations);

    printf("Do you wish to enter your own partial routes (y/n) [no]:
");
    gets(buffer);
    sscanf(buffer, "%c", &choice);

    if (choice == 'y' || choice == 'Y')
    {
        /* The user will enter partial routes */
        printf("Enter the number of partial routes: ");
        gets(buffer);
        sscanf(buffer, "%d", &pr);

        org = (int *) malloc(sizeof(int)*pr);
        dst = (int *) malloc(sizeof(int)*pr);

        for(i=0; i<pr; ++i)
        {
            printf("Enter the origin & destination of the partial route
"
                   "pair #%d: ", i+1);
            gets(buffer);
            sscanf(buffer, "%d %d", &org[i], &dst[i]);

            /* Check the validity of inputs */
            if (org[i] <= 0 || org[i] > stations ||
                dst[i] <= 0 || dst[i] > stations ||
                org[i] >= dst[i]) {
                printf("Invalid Station number (%d, %d). RE-ENTER
data\n",
```

```
                        org[i], dst[i]);
                /* Go back a step and continue */
                --i;
                continue;
            }
        }
    }
    else if (stations < 6)  {
        /* No partial routes possible */
        pr = 0;
    }
    else  {
        /* Generate DEFAULT partial routes */

        pr = NUM_DEFAULT_PR;

        org = (int *) malloc(sizeof(int)*pr);
        dst = (int *) malloc(sizeof(int)*pr);

        /* First Default Pair */
        org[0] = stations / 6;
        dst[0] = stations / 2;

        /* Second Default Pair */
        org[1] = stations / 3;
        dst[1] = 2 * stations / 3;

        /* Third Default Pair */
        org[2] = stations / 2;
        dst[2] = 5 * stations / 6;
    }

    /* Print all the partial routes */
    if (pr == 0) {
        printf("There are no partial routes\n");
    }
    for (i=0; i<pr; i++) {
        printf("Partial routes #%d: %d, %d\n", i+1, org[i], dst[i]);
    }

    /* Speed of the Bus*/
    printf("The Speed of the Bus is: %d mile(s)/min\n", speed);

    /* Headway of the bus at different times*/
    printf("Do you wish to enter your own headway times (y/n) [no]: ");
    gets(buffer);
    sscanf(buffer, "%c", &choice);

    if (choice == 'y' || choice == 'Y') {
        /* Enter the values of the headway for different times*/

        printf("Enter the values of the ampeak, pmpeak, midday and
offpeak: ");
        gets(buffer);
        sscanf(buffer, "%d %d %d %d", &ampeak, &pmpeak, &midday,
&offpeak);
    }
```

```
    else  {
        /* Default values of the headway are already assigned in the
         * beginning of the program */
    }

    printf("The ampeak, pmpeak, midday and offpeak values are: %d %d %d
%d\n",
            ampeak,pmpeak,midday,offpeak);

}
```

Appendix C.2: Computer Programs for Creating an Input File for the Entropy Maximization problem

Constraint Generator Program

```
#include <stdio.h>

/* Some basic type declarations */
#define TRUE            1
#define FALSE           0

#define MAX_STATIONS    100

/* Default Input File names */
#define DISTANCE_FILE           "distance.txt"
#define SAMPLE_IN_DEP_FILE      "departure.txt"
#define SAMPLE_IN_ARR_FILE      "arrival.txt"

/* Output file names */
#define DISTANCE_MATRIX         "distance.out"
#define SAMPLE_LHS_OUT_FILE     "data.a"
#define SAMPLE_RHS_OUT_FILE     "data.b"
#define SAMPLE_SIZE_OUT_FILE    "data.sizes"

/* Actual input file names */
char dist_file[80];
char in_dep_file[80];
char in_arr_file[80];
char out_lhs_file[80];
char out_rhs_file[80];
char out_size_file[80];

/* Forward declarations of functions */
void getInputFileNames(void);
void readDistFile(char filename[], float *dist_array, int stations);
void readDataFile(char filename[], int *array, int stations);
void generateDistMatrix(float *dist_matrix, float *dist_array, int
stations);
void writeDistFile(char filename[], float *dist_matrix, int stations);
void writeLhsConstraintFile(char lhs_file[], int stations, int onData);
void writeRhsConstraintFile(char rhs_file[], int *array, int stations,
int onData);
void writeAvgTripLenConstraint(char lhs_file[], char rhs_file[], float
avg,
                               float *matrix, int stations);
void writeDataSizes(char size_file[], int stations);


main(int argc, char **argv)
{
    int i;
    int num_stations;
    float avg_trip_len, min_trip_len, max_trip_len;

    char buffer[100];
```

```
    /* Input Data Arrays */
    float dist_array[MAX_STATIONS];
    int dep_data_array[MAX_STATIONS];
    int arr_data_array[MAX_STATIONS];

    /* Generated Matrix Data Structures */
    float dist_matrix[MAX_STATIONS][MAX_STATIONS];

    printf("\n");
    if (argc == 1) {
        do {
            printf("Enter the number of Stations: ");
            gets(buffer);
            sscanf(buffer, "%d", &num_stations);
        } while (strcmp(buffer, "") == 0 || num_stations <= 0);
    }
    else {
        printf("\nUSAGE: seperate_matrix\n\n");
        exit(0);
    }

    /* No need to run this program unless there are atleast 2 stations
*/
    if (num_stations < 2) {
        printf("Number of stations SHOULD be ATLEAST 2...\n\n");
        exit(0);
    }

    getInputFileNames();

    /* Read the Distance vector file */
    readDistFile(dist_file, dist_array, num_stations);

    /* Read all the DEPARTURE traffic data files */
    readDataFile(in_dep_file, dep_data_array, num_stations);
    /* Read all the ARRIVAL traffic data files */
    readDataFile(in_arr_file, arr_data_array, num_stations);

#if 0
    /* Ask the user, whether the data entered is ON/OFF data */
    printf("Is this ON data or OFF data (on/off) [on]: ");
    gets(buffer);
    if ((strcmp(buffer, "") == 0) || (strcmp(buffer, "off"))) {
        /* Consider the input data as "ON" traffic data */
        onData = TRUE;
    }
    else {
        /* Consider the input data as "OFF" traffic data */
        onData = FALSE;
    }
#endif
    /* Generate the distance matrix */
    generateDistMatrix((float *)dist_matrix, dist_array, num_stations);

    /* Get the Average trip length data from the user */
    min_trip_len = dist_matrix[0][1];
```

```
    max_trip_len = dist_matrix[0][num_stations-1];
    do {
        printf("Enter the average trip length (MIN %.1f - MAX %.1f)
[%.1f]: ",
                      min_trip_len,
                      max_trip_len,
                      (min_trip_len + max_trip_len) / 2);
        gets(buffer);
        if (strcmp(buffer, "") == 0) {
            avg_trip_len = (min_trip_len + max_trip_len) / 2;
        }
        else {
            sscanf(buffer, "%f", &avg_trip_len);
        }
    } while (avg_trip_len < min_trip_len || avg_trip_len >
max_trip_len);
    printf("\n");

    /* Write the generated distance matrix to output file */
    writeDistFile(DISTANCE_MATRIX, (float *)dist_matrix, num_stations);


    /* *** Generate LHS *** */
    /* Write LHS Departure Constraints to the output file */
    writeLhsConstraintFile(out_lhs_file, num_stations, TRUE);
    /* Write LHS Arrival Constraints to the output file */
    writeLhsConstraintFile(out_lhs_file, num_stations, FALSE);


    /* *** Write RHS *** */
    /* Write RHS Departure data */
    writeRhsConstraintFile(out_rhs_file, dep_data_array, num_stations,
TRUE);
    /* Write RHS Arrival data */
    writeRhsConstraintFile(out_rhs_file, arr_data_array, num_stations,
FALSE);


#if 0
    /* Write LHS and RHS of the AVG trip length constraint */
    writeAvgTripLenConstraint(out_lhs_file,
                              out_rhs_file,
                              avg_trip_len,
                              (float *)dist_matrix,
                              num_stations);

    printf("INFO: Wrote LHS Constraints to output file: %s\n",
out_lhs_file);
    printf("INFO: Wrote RHS Constraints to output file: %s\n",
out_rhs_file);
#endif

    /* Write the data size */
    writeDataSizes(out_size_file, num_stations);
    printf("INFO: Wrote M,N data sizes to output file: %s\n",
out_size_file);
```

```
    printf("\n");

}
void getInputFileNames(void)
{
    char buffer[100];

    /* Get the Distance vector file name from the user */
    printf("Enter the file_name for Distance between Stations [%s]: ",
                                              DISTANCE_FILE);
    gets(buffer);
    sscanf(buffer, "%s", dist_file);
    if (strcmp(dist_file, "") == 0) {
        memcpy(dist_file, DISTANCE_FILE, strlen(DISTANCE_FILE));
    }

    /* Get the Departure Data file name from the user */
    printf("Enter the input file name for Departure Data [%s]: ",
                                       SAMPLE_IN_DEP_FILE);
    gets(buffer);
    sscanf(buffer, "%s", in_dep_file);
    if (strcmp(in_dep_file, "") == 0) {
        memcpy(in_dep_file, SAMPLE_IN_DEP_FILE,
strlen(SAMPLE_IN_DEP_FILE));
    }

    /* Get the Arrival Data file name from the user */
    printf("Enter the input file name for Arrival Data [%s]: ",
                                       SAMPLE_IN_ARR_FILE);
    gets(buffer);
    sscanf(buffer, "%s", in_arr_file);
    if (strcmp(in_arr_file, "") == 0) {
        memcpy(in_arr_file, SAMPLE_IN_ARR_FILE,
strlen(SAMPLE_IN_ARR_FILE));
    }

    printf("Enter the output file name for LHS [%s]: ",
SAMPLE_LHS_OUT_FILE);
    gets(buffer);
    sscanf(buffer, "%s", out_lhs_file);
    if (strcmp(out_lhs_file, "") == 0) {
        memcpy(out_lhs_file, SAMPLE_LHS_OUT_FILE,
strlen(SAMPLE_LHS_OUT_FILE));
    }

    printf("Enter the output file name for RHS [%s]: ",
SAMPLE_RHS_OUT_FILE);
    gets(buffer);
    sscanf(buffer, "%s", out_rhs_file);
    if (strcmp(out_rhs_file, "") == 0) {
        memcpy(out_rhs_file, SAMPLE_RHS_OUT_FILE,
strlen(SAMPLE_RHS_OUT_FILE));
    }

    printf("Enter the output file name for sizes [%s]:
",SAMPLE_SIZE_OUT_FILE);
    gets(buffer);
```

```
    sscanf(buffer, "%s", out_size_file);
    if (strcmp(out_size_file, "") == 0) {

memcpy(out_size_file,SAMPLE_SIZE_OUT_FILE,strlen(SAMPLE_SIZE_OUT_FILE))
;
    }

    printf("\n");
}




void readDistFile(char filename[], float *array, int stations)
{
    int i;
    FILE *fp;

    if ((fp = fopen(filename, "r")) == NULL) {
        printf("\nERROR: Couldn't open Input file: %s\n\n", filename);
        exit(0);
    }

    /* Distance from station to itself is zero */
    array[0] = 0.0;

    i = 1;
    while (i < stations && fscanf(fp, "%f\n", &array[i]) != EOF) {
        i++;
    }
    fclose(fp);

    if (i != stations) {
        printf("\nERROR: Missing Data. Tried to read %d entries, "
               "but read only %d entries.\n\n", stations, i);
        exit(0);
    }
}



void readDataFile(char filename[], int *array, int stations)
{
    int i;
    FILE *fp;

    if ((fp = fopen(filename, "r")) == NULL) {
        printf("\nERROR: Couldn't open Input file: %s\n\n", filename);
        exit(0);
    }

    i = 0;
    while (i < stations && fscanf(fp, "%d\n", &array[i]) != EOF) {
        i++;
    }
    fclose(fp);
```

```
    if (i != stations) {
        printf("\nERROR: Missing Data. Tried to read %d entries, "
               "but read only %d entries.\n\n", stations, i);
        exit(0);
    }

#if 0
    /* Reset the passangers boarding at END station to ZERO */
    array[stations - 1]  = 0;
#endif
}



void generateDistMatrix(float *matrix, float *array, int stations)
{
    int dst, org;

    for (org=0; org<stations; org++) {
        for (dst=0; dst<stations; dst++) {
            if (org > dst) {
                /* We are not interested in reverse distance */
                matrix[org*MAX_STATIONS + dst] = 0;
            }
            else if (org == dst) {
                /* Distance to oneself is ZERO */
                /* Example: Distance from Station "3" to Stations "3" =
0 */
                matrix[org*MAX_STATIONS + dst] = 0;
            }
            else {
                /* Dist(1,5) = Dist(1,4) + Dist(4,5)
                 * Dist(1,4) has already been calculated and is part of
matrix
                 * Dist(4,5) comes from Data of Distance array file =
Dist(5)
                 */
                matrix[org*MAX_STATIONS + dst] =
                    matrix[org*MAX_STATIONS + dst - 1] + array[dst];
            }
        }
    }
}

void writeDistFile(char filename[], float *matrix, int stations)
{
    FILE *fp;
    int col, row;

    if ((fp = fopen(filename, "w")) == NULL) {
        printf("\nERROR: Couldn't open Output file: %s\n\n", filename);
        exit(0);
    }

    for (row=0; row<stations; row++) {
        for (col=0; col<stations; col++) {
            fprintf(fp, "%.1f ", matrix[row*MAX_STATIONS+col]);
```

```
        }
        fprintf(fp, "\n");
    }

    fclose(fp);

    printf("INFO: Wrote Distance to output file: %s\n", filename);
}

void writeLhsConstraintFile(char lhs_file[], int stations, int onData)
{
    FILE *fp_lhs;
    int col, row, line;
    char mode[10];

    if (onData) {
        strcpy(mode, "w");
    }
    else {
        strcpy(mode, "a");
    }

    if ((fp_lhs = fopen(lhs_file, mode)) == NULL) {
        printf("\nERROR: Couldn't open Output LHS file: %s\n\n",
lhs_file);
        exit(0);
    }

    for (line=0; line<stations-1; line++) {

        /* Generate the LHS of the equation */
        for (row=0; row<stations-1; row++) {
            for (col=row+1; col<stations; col++) {

                if (onData) {
                    if (line == row) {
                        fprintf(fp_lhs, "%d ", 1);
                    }
                    else {
                        fprintf(fp_lhs, "%d ", 0);
                    }
                }
                else {
                    if (line+1 == col) {
                        fprintf(fp_lhs, "%d ", 1);
                    }
                    else {
                        fprintf(fp_lhs, "%d ", 0);
                    }
                }

            }
        }
        fprintf(fp_lhs, "\n");
    }

    fclose(fp_lhs);
```

```
}

void writeRhsConstraintFile(char rhs_file[], int *array,
                            int stations, int onData)
{
    FILE *fp_rhs;
    int line;
    char mode[10];
    int start, end;

    if (onData) {
        strcpy(mode, "w");
        start = 0;
        end   = stations-1;
    }
    else {
        strcpy(mode, "a");
        start = 1;
        end   = stations;
    }

    if ((fp_rhs = fopen(rhs_file, mode)) == NULL) {
        printf("\nERROR: Couldn't open Output RHS file: %s\n\n",
rhs_file);
        exit(0);
    }

    for (line=start; line<end; line++) {
        /* Generate the RHS of the equation */
        fprintf(fp_rhs, "%d\n", array[line]);
    }

    fclose(fp_rhs);
}

#if 0
void writeAvgTripLenConstraint(char lhs_file[], char rhs_file[],
                               float avg, float *matrix, int stations)
{
    FILE *lhs_fp, *rhs_fp;
    int dst, org;

    if ((lhs_fp = fopen(lhs_file, "a")) == NULL) {
        printf("\nERROR: Couldn't open Output file: %s\n\n", lhs_file);
        exit(0);
    }
    if ((rhs_fp = fopen(rhs_file, "a")) == NULL) {
        printf("\nERROR: Couldn't open Output file: %s\n\n", rhs_file);
        exit(0);
    }

    /* Generate the LHS of the equation */
    for (org=0; org<stations; org++) {
        for (dst=0; dst<stations; dst++) {
            if (org > dst) {
                /* We are not interested in reverse distance */
            }
```

```
            else if (org == dst) {
                /* Distance to oneself is ZERO */
                /* Example: Distance from Station "3" to Stations "3" =
0 */
            }
            else {
                fprintf(lhs_fp,"%.1f ", (matrix[org*MAX_STATIONS + dst]
- avg));
            }
        }
    }
    fprintf(lhs_fp, "\n");

    /* Generate the RHS of the equation */
    fprintf(rhs_fp, "%d\n", 0);

    fclose(lhs_fp);
    fclose(rhs_fp);
}
#endif



void writeDataSizes(char size_file[], int stations)
{
    FILE *fp;
    int constraints = 0, variables = 0;

#if 0
    constraints = (2 * stations) - 1;
#endif
    constraints = (2 * stations) - 2;
    variables   = stations * (stations -1) / 2;

    if ((fp = fopen(size_file, "w")) == NULL) {
        printf("Couldn't write to %s\n", size_file);
        exit(0);
    }
    fprintf(fp, "%d,%d\n", constraints, variables);
    fclose(fp);
}
```

Appendix C.3: Computer Programs for Entropy Maximization

The solver consists of 12 programs, one in C and the other in FORTRAN.

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
c  Master Control Program
c
      call c_control
      end


/*
   C CONTROL

   This module dynamically allocates memory space for all internal
   tables and arrays.

*/

#define null 0

#include <stdio.h>
struct sizes {
              int m;
              int n;
};

extern struct sizes sizes_;

struct perturbation {
              double mu;
};

extern struct perturbation perturbation_;

void c_control__()
{
int  m, n; /*number of primal constraints and primal variables */
double mu; /*perturbation */
double *c, *b; /*array for primal objective and r.h.s.  */
double *x; /*array for dual variables */
double *a; /*double array for matrix A */
double *at; /*double array for matrix A transpose */
double *g; /*array for gradient  */
double *H; /*double array for matrix Hessian */
double *Hi; /*double array for the inverse of matrix Hessian */
double *HH; /*working double array for matrix inversion */
double *v; /*array for working variable v  */
double *w1; /*array for working variable w1  */
double *d, *z; /*array for direction d and direction z  */
double *x0; /*array for initial solution for (dual) variables */
double *x_o; /*array for optimal solution for (dual) variables */
double *x_p; /*array for previous solution for (dual) variables */
double *y; /*array for primal variable y  */
```

```
int   rc, i;

/*
    Call a fortran subroutine to get array & matrix sizes: n,m
*/
        read_sizes_perturbation__();

        m = sizes_.m;
        n = sizes_.n;
        mu = perturbation_.mu;

/*  for debug, may be deleted to line marked !!!                        */
/*        printf("n = %d\n",n);                                         */
/*        printf("m = %d\n",m);                                         */
/*  !!!                                                                 */
/* allocate for x, p & c arrays                                         */
/* x(n), p(m) and c(n)                                                  */

if ( ((x  = (double *) calloc ( (m) , sizeof(double))) == null)         ||
     ((c  = (double *) calloc ( (n) , sizeof(double))) == null)         ||
     ((b  = (double *) calloc ( (m) , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating x, c & b arrays.\n\n");
      fflush(stdout);
      goto endc_control;
}


/*  for debug, may be deleted to line marked !!!                        */
/*      printf("rc     = %d\n",rc);                                     */
/*  !!!                                                                 */

/* allocate for g arrays                                                */
/* g(n)                                                                 */

if ( ((g  = (double *) calloc ( (m) , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating g arrays.\n\n");
      fflush(stdout);
      goto endc_control;
}

/*  !!!                                                                 */
/* allocate for double array (matrix) a and at                          */
/* at(n,m)                                                              */
if ( ((a  = (double *) calloc ( (m*n) , sizeof(double))) == null)       ||
     ((at = (double *) calloc ( (m*n) , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating a and at arrays.\n\n");
      fflush(stdout);
      goto endc_control;
```

```
}


/*  for debug, may be deleted to line marked !!!         */
/*      printf("rc     = %d\n",rc);                      */
/*  !!!                                                  */

/*  !!!                                                                */
/* allocate for double array (matrix) H and Hi                        */
/* H(m,m), Hi(m,m)                                                     */
if ( ((H  = (double *) calloc ( (m*m)  , sizeof(double))) == null)      ||
     ((Hi = (double *) calloc ( (m*m)  , sizeof(double))) == null)      ||
     ((HH = (double *) calloc ( (m*2*m)  , sizeof(double))) == null) )

{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating H and Hi matrices.\n\n");
      fflush(stdout);
      goto endc_control;
}


/*  for debug, may be deleted to line marked !!!         */
/*      printf("rc     = %d\n",rc);                      */
/*  !!!                                                  */

/*  !!!                                                                */
/* allocate for d, z & v arrays                                        */
/* d(m), z(m), v(m) and w1(n)                                          */

if ( ((d  = (double *) calloc ( (m)  , sizeof(double))) == null)        ||
     ((z  = (double *) calloc ( (m)  , sizeof(double))) == null)        ||
     ((v  = (double *) calloc ( (m)  , sizeof(double))) == null)        ||
     ((w1 = (double *) calloc ( (n)  , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating d, z, v & w1 arrays.\n\n");
      fflush(stdout);
      goto endc_control;
}

/*  !!!                                                                */
/* allocate for x0, x_o arrays                                         */
/* x0(m), x_o                                                          */

if ( ((x0  = (double *) calloc ( (m)  , sizeof(double))) == null)       ||
     ((x_o = (double *) calloc ( (m)  , sizeof(double))) == null)       ||
     ((x_p = (double *) calloc ( (m)  , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating x0, x_o & x_p arrays.\n\n");
      fflush(stdout);
      goto endc_control;
}
```

```
/*  for debug, may be deleted to line marked !!!                        */
/*      printf("rc     = %d\n",rc);                          */
/*  !!!                                                                 */

if ( ((y  = (double *) calloc ( (n) , sizeof(double))) == null) )
{
      rc = 84;
      printf("\nFATAL ERROR in c_main.c.  rc = %d.\n\n",rc);
      printf("error in allocating y arrays.\n\n");
      fflush(stdout);
      goto endc_control;
}

/*  for debug, may be deleted to line marked !!!                        */
/*      printf("rc     = %d\n",rc);                          */
/*  !!!                                                                 */


/* now call the fortran driver                                      */

      f_control__( x, c, a, at, b, x0, g, H, Hi, HH, d, z, v, w1, x_o, x_p,
y);

endc_control:
      exit(rc);


}     /* end of c_control.c */
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
C/*******************************************************************/
C/*     Read the file data.sizes to obtain array.matrix sizes:
C/*     m and n, sizes of (dual) variable array and (primal) prior array.
C/*     The maximum sizes are 10**10 (10 digits).
C/*******************************************************************/
        subroutine read_sizes_perturbation

        implicit none
        Integer          m, n
        Double precision mu

        common /sizes/ m,n
        common /perturbation/ mu

        open (1, status='old', file='data.sizes')
        read (1,*) m,n
        close(1)

        open (2, status='old', file='data.perturbation')
        read (2,*) mu
         close(2)
```

```
        return
        end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine f_control( x, c, a, at, b, x0, g, H, Hi, HH, d, z, v,
     1                        w1, x_o, x_p, y)

c*******************************************************************
c     This program control the fortran subroutines
c*******************************************************************

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program
c
        Double precision  x(m), c(n), b(m), x0(m)
        Double precision  a(m,n), at(n,m)
        Double precision  g(m)
        Double precision  H(m,m), Hi(m,m), HH(m,2*m)
        Double precision  d(m), z(m), v(m), w1(n)
        Double precision  x_o(m), x_p(m)
        Double precision  y(n)
c
c     Local variables
c
        Integer           i, j, k
c
c***********************************************************************
c     The program instructions begin here.
c     This program (1) defines the optimization problem by reading
c                      data files.
c                  (2) calls the LE control, which is the main program for
c                      Curved Search Algorithm
c************************************************************************

c********************************************************************
c     Defines the optimization problem by reading data files:
c     data.x, data.p, data.bt and data.c.
c
c     NOTE: read matrix(1:m,1:n) reads columnwise, i.e. it fills in the
c           first column and then the second and so on.
c           Therefore, the data should be read into the transpose of
c           the original matrix and transpose it back to the original one
c           the data is stored rowwise, which is usually the case.
```

```
c*************************************************************************

        call ls_gen (a,b,c,x0,x_o)

c************************************************************************
c      Call the LE control routine.
c************************************************************************

        call le_control (x, c, a, at, b, x0, g, H, Hi, HH, d, z, v,
     1                   w1, x_o, x_p, y)

        return
      end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine ls_gen (a,b,c,x0,x_o)

        Implicit none

        integer m,n
        common /sizes/ m,n
        Double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program
c
        integer i,j
        Double precision  w

        Double precision  c(n), b(m), x0(m)
        Double precision  a(m,n)
        Double precision  x_o(m)

      open(12, FILE = 'data.a', STATUS = 'old')
      open(13, FILE = 'data.b', STATUS = 'old')
*     open(14, FILE = 'data.c', STATUS = 'old')

      read(12,*) ((a(i,j),j=1,n),i=1,m)
      read(13,*) (b(i),i=1,m)
*     read(14,*) (c(j),j=1,n)

      close(12)
      close(13)
*     close(14)

      w = 0.0
      do 400 i = 1, n
         c(i) = w
400   continue

      w = 0.0
```

```
      do 500 i = 1, m
         x0(i) = w
500   continue

      w = 1.0
      do 600 i = 1, m
         x_o(i) = w
600   continue


c      write(*,*) 'm  = # primal constraints   = ',m
c      write(*,*) 'n  = # primal variables     = ',n
c      write(*,*) 'mu                          = ',mu
c      write(*,*) 'a(1,1) , a(1,2), a(1,3)     = ',a(1,1),a(1,2),a(1,3)
c      write(*,*) 'a(2,1) , a(2,2), a(2,3)     = ',a(2,1),a(2,2),a(2,3)
c      write(*,*) ' '
c      write(*,*) 'a(m,1) , a(m,2), a(m,3)     = ',a(m,1),a(m,2),a(m,3)
c      write(*,*) ' '
c      write(*,*) 'b(1) , b(2), b(3)           = ',b(1),b(2),b(3)

      return
      end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
        subroutine le_control(x, c, a, at, b, x0, g, H, Hi, HH, d, z, v,
     1                        w1, x_o, x_p, y)
c***************************************************************************
c     This program control the CS subroutines.
c***************************************************************************
        Implicit none
        integer m,n
        common /sizes/ m,n
        Double precision  mu
        common /perturbation/ mu
c     The following arrays are allocated by the c_control.c program
        Double precision  x(m), c(n), b(m), x0(m)
        Double precision  a(m,n), at(n,m)
        Double precision  g(m)
        Double precision  H(m,m), Hi(m,m), HH(m,2*m)
        Double precision  d(m), z(m), v(m), w1(n)
        Double precision  x_o(m), x_p(m)
        Double precision  y(n)
c     The following variables are initially defined in this program
c     and may be passed to some of its subroutines.

        Double precision  epsilon_g, epsilon_l, sigma, gamma
        Double precision  alpha, beta
        Double precision  tmin, tmax, t
        Double precision  value_tmin, value_tmax, value_t
        Double precision  value, current_value
        Double precision  cpu_g_h_exp, cpu_g_h_non_exp, cpu_inversion
```

```
c     Local variables
      Double precision    w, ww, gvk, gammak, gk2, gk, vk
      Double precision    dk, dk1, r1, r2
      Integer             IPOS, i, j, k, k_rescaled
      Real                ustime(2), etime
      Real                time_0, time_1, time_2, time_3, time_4
c**********************************************************************
c     The program instructions begin here.
c**********************************************************************
c**********************************************************************
c      LE Control:     Initialization
c**********************************************************************
       epsilon_g = 1.0d-5
       epsilon_l = 1.0d-10
       sigma   = 1.0d-5
       gamma   = 1.0d-5
       alpha   = 1.0
       beta    = 1.0
       cpu_g_h_exp     = 0.0d+00
       cpu_g_h_non_exp = 0.0d+00
       cpu_inversion   = 0.0d+00
       time_0 = etime(ustime)
       time_1 = time_0
       time_2 = time_0
c      write(*,*) 'The test problem:'
c      write(*,*) 'Perturbation parameter mu    = ', mu
c      write(*,*) 'Number of primal constraints = ', m
c      write(*,*) 'Number of primal variables   = ', n
c      write(*,*) 'Objective vector:'
c      write(*,*) 'c1-3=', c(1),c(2),c(3)
c      write(*,*) 'cn=', c(n-2),c(n-1),c(n)
c      write(*,*) 'Matrix:'
c      write(*,*) 'r1: 1,1:3=', a(1,1),a(1,2),a(1,3)
c      write(*,*) '    1,4:6=', a(1,4),a(1,5),a(1,6)
c      write(*,*) '    1,7:9=', a(1,7),a(1,8),a(1,9)
c      write(*,*) '    11:12=', a(1,10),a(1,11),a(1,12)
c      write(*,*) 'rm-1:,1:3=', a(m-1,1), a(m-1,2), a(m-1,3)
c      write(*,*) ' m-1,4:6=', a(m-1,4), a(m-1,5), a(m-1,6)
c      write(*,*) ' m-1,7:9=', a(m-1,7), a(m-1,8), a(m-1,9)
c      write(*,*) '    11:12=', a(m-1,10), a(m-1,11), a(m-1,12)

c      write(*,*) 'rm-1:..n=', a(m-1,n-2), a(m-1,n-1), a(m-1,n)

c      write(*,*) 'rm :, 1:3=', a(m,1), a(m,2), a(m,3)
c      write(*,*) '      m,4:6=', a(m,4), a(m,5), a(m,6)
c      write(*,*) '    m,7:9=', a(m,7), a(m,8), a(m,9)
c      write(*,*) '    11:12=', a(m,10), a(m,11), a(m,12)
c      write(*,*) 'rm   :..n=', a(m,n-2), a(m,n-1), a(m,n)
c      write(*,*) 'The constants (+ r.h.s.):'
c      write(*,*) 'b1-3=', b(1),b(2),b(3)
c      write(*,*) 'b4-6=', b(4), b(5), b(6)
c      write(*,*) 'b7-8=', b(7), b(8)
c      write(*,*) 'bn=', b(m-2),b(m-1),b(m)
c      write(*,*) 'Epsilon for |gradient| check           = ', epsilon_g
c      write(*,*) 'Epsilon for line search stop (initial) = ', epsilon_l
```

```
       do 10 i=1,m
          x(i)=x0(i)
          v(i)=x0(i)
10    continue !i
c       do i = 1, m
c            x_p(i) = x(i)
c       enddo ! i

c      write(*,*) 'Initial solution:'
c      write(*,*) 'x01-3=', (x(k),k=1,3)
c      write(*,*) 'x04-6=', (x(k),k=4,6)
c      write(*,*) 'x07-8=', (x(k),k=7,8)
      call objective(v, c, a, b, value)
      current_value = value
      write(*,*) 'Initial objective value = ', value
31    format('x*1= ',d20.14)
32    format('x*2= ',d20.14)
33    format('x*3= ',d20.14)
34    format('x*4= ',d20.14)
35    format('x*5= ',d20.14)
c     The first 5 elements of the optimal vector
      write(*,*) 'Optimal solution for convergence rate calculation:'
      write (*,*) 'The first 5 elements of the optimal vector:'
      do i = 1, 5
           write(*,*) x_o(i)
      enddo
c     for each of the n dual LP constraints, find the slack:
      write(*,*) 'The dual LP constraint violations (>0=violation)'
      do j = 1, n
           w = 0.0d+0
           do i = 1, m
                w = w + a(i,j) * x(i)
           enddo !i
           if ( w .gt. 10.0)  then
                 write (*,*) 'ABNORMAL VIOLATION OF LP DUAL CONSTRAINT'
           endif
d          write (*,*) j, w - c(j)
      enddo !j
      w = 0.0d+0
       do i = 1, m
            w = w + (x(i) - x_o(i)) * (x(i) - x_o(i))
       enddo !i
       dk1 = sqrt(w)

c*********************************************************************
c      We now start iteration.
c*********************************************************************

        k = 0
100        k = k + 1

        write(*,*) '*************************'
        write(*,*) 'ITERATION k = ',
c*********************************************************************
c       Calculate gradient and Hessain together for computational efficiency.
c*********************************************************************
d       write(*,*) 'before grad_hessain'
```

```
        time_3 = etime(ustime)
        call grad_hessian( x, c, a, b, g, H, w1, cpu_g_h_exp,
     1                                          cpu_g_h_non_exp)
        time_4 = etime(ustime)
d       write(*,*) 'grad and Hessian cpu = ', time_4 - time_3
c**********************************************************************
c       Check optimality
c*********************************************************************

        w = 0.0d+0
        do i = 1, m
             w = w + g(i) * g(i)
        enddo !i
        gk2 = w
        gk = sqrt(gk2)
        if (gk .lt. epsilon_g) then
           write(*,*) 'OPTIMALITY REACHED!'
           write(*,*) 'It was reached when the norm of the gradient '
           write(*,*) '     reduced to within ', epsilon_g, ' of 0.'
           go to 1000
        else
           write(*,*) 'not optimal yet! gk = ', gk
        endif
c**********************************************************************
c       Solve Hv=g, i.e. calculate v=Hig.
c       For now, implement the matrix inversion for future use.
c       If the matrix is not invertible, use the steepest decent direction.
c**********************************************************************
d       write(*,*) 'before inverse'
        time_3 = etime(ustime)
        call inverse(H, Hi, HH, IPOS)
        time_4 = etime(ustime)
d       write(*,*) 'matrix inversion cpu = ', time_4 - time_3
        cpu_inversion = cpu_inversion + time_4 - time_3
        if (IPOS .eq. 1)  then
         write(*,*) 'inversion completed'
           go to 200
        else
           if (IPOS .eq. 0)  then
            write(*,*) 'INVERSION NOT COMPLETED!!!'
              do i = 1, m
                   d(i) = - g(i)
                   z(i) = 0.0d+0
              enddo
              write(*,*) 'steepest descent direction used!'
              go to 2000
           else
            write(*,*) 'inversion completed'
            write(*,*) 'H not positive definite'
              go to 200
           endif
         endif
200     continue
c**********************************************************************
c       Calculate variables:
c       Note that the determinant sigmak is not being calculated yet.
c**********************************************************************
```

```
d       write(*,*) 'before misc. calculation'
        time_3 = etime(ustime)
        do i = 1, m
             v(i) = 0.0d+00
             do j = 1,m
                  v(i) = v(i) + Hi(i,j) * g(j)
             enddo ! j
        enddo !i

        w = 0.0d+0
        do i = 1, m
             w = w + g(i) * v(i)
        enddo !i
        gvk = w
c  Note that gk2 and gk were obtained earlier!
c       gk2 = dotproduct(g,g)
c       gk = sqrt(gk2)
        w = 0.0d+0
        do i = 1, m
             w = w + v(i) * v(i)
        enddo !i
        vk = sqrt(w)
        gammak = gvk / (gk2 * vk)
        if (gammak .lt. 0.0d+00) then
            gammak = - gammak
        endif
c***********************************************************************
c       Determine step type and actual direction.
c       Note that sigmak is not being checked yet.
c***********************************************************************
d       write(*,*) 'before direction determination'
        if (gammak .gt. gamma) then
           do i = 1, m
                d(i) = - beta * gk2 * v(i) / gvk
                z(i) = - alpha * gk * g(i)
           enddo
        else
           do i = 1, m
                d(i) = - g(i)
                z(i) = 0.0d+0
           enddo
           write(*,*) 'steepest descent direction used!'
        endif
c***********************************************************************
c       Find a three point pattern: the interval (tmin,tmax) will contain t*.
c**********************************************************************
d       write(*,*) 'before three point determination'
2000    time_3 = etime(ustime)
        call interval(x, c, a, b, d, z, v, current_value,
     1                tmin, value_tmin, tmax, value_tmax, k)
        time_4 = etime(ustime)
        write(*,*) 'tmin, tmax = ', tmin, tmax
        if (tmin .gt. tmax) then
           write(*,*) 'NONO'
        endif
d       write(*,*) 'three-point determ cpu = ', time_4 - time_3
```

```
c***********************************************************************
c       Line Search: Golden-Section Search.
c***********************************************************************
d       write(*,*) 'before golden section search'
        time_3 = etime(ustime)
        call golden( x, c, a, b, d, z, v, tmin, value_tmin, tmax,
     1              value_tmax, t, value_t, epsilon_l)
        current_value = value_t
c  Note that value_t is the value at the optimal step
c  but x contains the old solution.
c  x will be updated later.
        time_4 = etime(ustime)
        write(*,*) 'tmin, t, tmax = '
        write(*,*) tmin, t, tmax
        write(*,*) value_tmin, value_t, value_tmax
d       write(*,*) 'golden section search cpu = ', time_4 - time_3
c**********************************************************************
c       Iterate
c**********************************************************************
        If (t .eq. 0.0d+0) then
           write(*,*) 'LINE SEARCH ENDED AT THE ORIGIN, i.e. t = 0'
           call grad_hessian( x, c, a, b, g, H, w1,
     1                       cpu_g_h_exp, cpu_g_h_non_exp)
           w = 0.0d+0
           do i = 1, m
                w = w + g(i) * g(i)
           enddo !i
           gk2 = w
           gk = sqrt(gk2)
           go to 1000
        endif
        do i = 1, m
             x(i) = x(i) + t * d(i) + 0.5 * t * t * z(i)
        enddo ! i
c       check which LP dual constrints
c       are being violated
c     for each of the n dual LP constraints, find the slack:
      write(*,*) 'The dual LP constraint violations (>0=violation)'
      do j = 1, n
           w = 0.0d+0
           do i = 1, m
                w = w + a(i,j) * x(i)
           enddo !i
           if ( w .gt. 10.0)  then
                 write (*,*) 'ABNORMAL VIOLATION OF LP DUAL CONSTRAINT'
           endif
d          write (*,*) j, w - c(j)
      enddo !j
        w = 0.0
        do i = 1, m
             ww =  t * d(i) + 0.5 * t * t * z(i)
             w  = w + ww * ww
        enddo ! i
        w = sqrt(w)
        write (*,*) 'The distance between xk and xk-1 = ', w
d       write(*,*) 'x1-3=', x(1:3)
d       write(*,*) 'x4-5=', x(4:5)
```

```
        dk = dk1
        w = 0.0d+0
        do i = 1, m
             w = w + (x(i) - x_o(i)) * (x(i) - x_o(i))
        enddo !i
        dk1 = sqrt(w)
        r1 = dk1 / dk
        r2 = dk1 / (dk * dk)
        write(*,*) 'Objective valje = ', current_value
        write(*,*) 'dk = ', dk, '; dk1 = ',dk1
        write(*,*) 'dk1/dk = ', r1, '; dk1/(dk*dk) = ',r2
        time_1 = time_2
        time_2 = etime(ustime)
        write(*,*) 'cpu time for this iteration = ',time_2 - time_1
        go to 100
51      format('x1= ',d20.14)
52      format('x2= ',d20.14)
53      format('x3= ',d20.14)
54      format('x4= ',d20.14)
55      format('x5= ',d20.14)
1000    write(*,*) '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
        write(*,*) 'TOTAL # OF ITERATIONS: ', k
        write(*,*) 'Optimal solution reached!'
        write(*,*) 'g1-3=', g(1),g(2),g(3)
        write(*,*) 'g4-6=', g(4),g(5),g(6)
        write(*,*) 'gk = ', gk
        write(*,51) x(1)
        write(*,52) x(2)
        write(*,53) x(3)
        write(*,54) x(4)
        write(*,55) x(5)
c       Temperarily give up on creating a file for optimal solutions
c       open(11, FILE = 'data.x_o', STATUS = 'NEW')
c       Print out the first 10 elements of the dual optimal solution.
        write(*,*) 'First 10 elements of the dual optimal solution:'
        do i = 1, 10
             write(*,*) x(i)
        enddo !i
        call objective(x, c, a, b, value)
        write(*,*) 'optimal objective value = ', value
c       NOTE: should never use a k loop within a k loop
c             It is dangerous
        do j = 1, n
           w = 0.0d+0
           do i = 1, m
                w = w + a(i,j) * x(i)
           enddo !i
           w = ((w - c(j)) / mu) - 1
           y(j) = exp (w)
        enddo !j
        write(*,*) 'Optimal primal solution reached!'
        write(*,*) 'y1-3=', y(1),y(2),y(3)
        write(*,*) 'y4-6=', y(4),y(5),y(6)
        write(*,*) 'y7-9=', y(7),y(8),y(9)
        write(*,*) 'y..n-6=', y(n-8),y(n-7),y(n-6)
        write(*,*) 'y..n-3=', y(n-5),y(n-4),y(n-3)
        write(*,*) 'y..n=', y(n-2),y(n-1),y(n)
```

```
         w = 0.0
         do j = 1, n
          w=w+y(j)
        enddo !j
        write(*,*) 'Now, the total number of trips =!', w
        write(*,*) 'Now, check if y satisfies the prescribed sums!'
        w = 0.0
        do j = 1, n
           w = w + c(j) * y(j)
        enddo! j
        write(*,*) 'LP PRIMAL OBJECTIVE value = ', w
c   First print out the first 8 constraints:
        do i = 1, 8
             w = 0.0d+0
             do j = 1, n
                  w =  w + a(i,j) * y(j)
             enddo !j
             write(*,*) i, ' th P constraint: ', w, '=?', b(i)
        enddo !i

c   Now, print out the last 8 constraints:
        do i = m-7, m
             w = 0.0d+0
             do j = 1, n
                  w =  w + a(i,j) * y(j)
             enddo !j
             write(*,*) i, ' th P constraint: ', w, '=?', b(i)
        enddo !I
      open(19, FILE = 'data.od', STATUS = 'new')
      write(19,*) (y(j),j=1,n)
      close(19)

        time_2 = etime(ustime)
        write(*,*) 'Total cpu on hessian (expo) = ', cpu_g_h_exp
        write(*,*) 'Total cpu on hessian (rest) = ', cpu_g_h_non_exp
        write(*,*) '  '
        w = cpu_g_h_exp + cpu_g_h_non_exp
        write(*,*) 'Total cpu on hessian        = ', w
        write(*,*) 'Total cpu on inversion      = ', cpu_inversion
        w = time_2 - time_0 - cpu_g_h_exp -
     1      cpu_g_h_non_exp - cpu_inversion
        write(*,*) 'Total cpu on all the rest   = ', w
        write(*,*) '  '
        write(*,*) 'TOTAL CPU                   = ', time_2 - time_0

9000    return
      end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
        subroutine objective(v, c, a, b, value)
```

```
c************************************************************************
c     This program control the CS subroutines
c************************************************************************

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  c(n), b(m)
        Double precision  a(m,n)
        Double precision  v(m)

c     Local variables

        Double precision value, w
        Integer          i, j, k

c*****************************************************************
c     The instructions begin here.
c     Note that pass only the working array to "objective".
c*****************************************************************

      value = 0.0d+0
      do j = 1, n
           w = 0.0d+0
           do i = 1, m
                w = w + a(i,j) * v(i)
           enddo !i
           w = ((w - c(j)) / mu ) - 1.0

c      warning if w exceeds 300 for possible overflow

           if (w .gt. 300.0) then
              write(*,*) 'POSSIBLE OVERFLOW!  exponent = ', w
           endif

           value = value + exp (w)
       enddo !j

       w = 0.0d+0
       do i = 1, m
            w = w + b(i) * v(i)
       enddo !i

       value = -1.0 * (w - mu * value)

       return
       end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
        subroutine  grad_hessian( x, c, a, b, g, H, w1,
     1                           cpu_g_h_exp, cpu_g_h_non_exp)
c****************************************************************************
c     This program calculates the gradient and Hessian
c     of the dual objective function.  The reason for combination is
c     to capitalize on common calculations.
c     v is used as a working array here.
c****************************************************************************

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision  mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  x(m), c(n), b(m)
        Double precision  a(m,n)
        Double precision  g(m)
        Double precision  H(m,m)
        Double precision  w1(n)

c      Variables passed by cs_control

        Double precision cpu_g_h_exp, cpu_g_h_non_exp

c      Local variables

        Double precision  w, sg, sH
        Integer           i, j, k, l, k1, k2

        Real                ustime(2), etime
        Real                time_5, time_6, time_7
        Real                time_15, time_16, time_17


c****************************************************************************
c     The program instruction starts here.
c     It calculates both the gradient and the Hessian Matrix
c     The first do loop calcualtes an array of numbers to be used by
c     both the gradient and Hessian matrix.
c****************************************************************************

        time_5 = etime(ustime)
        time_15 = ustime(1)

        do j = 1,n
             w = 0.0d+0
             do i = 1,m
                  w = w + a(i,j) * x(i)
```

```
              enddo !i
              w = ( (w - c(j)) / mu ) - 1.0
              w1(j) = exp (w)
         enddo !j

         time_6 = etime(ustime)
         time_16 = ustime(1)
         write(*,*) 'exp function calculations cpu = ', time_6 - time_5
         write(*,*) 'calculations cpu = ', time_16 - time_15

         cpu_g_h_exp = cpu_g_h_exp + time_6 - time_5

         do k = 1,m
              sg = 0.0d+0
              do j = 1,n
                   sg = sg + w1(j) * a(k,j)
              enddo !j
              g(k) = sg - b(k)
         enddo !k

         do k1 = 1, m
              do k2 = 1, k1
              sH = 0.0d+0
                   do j = 1, n
                        sH = sH + w1(j) * a(k1,j) * a(k2,j)
                   enddo !j
              H(k1,k2) = sH / mu
              enddo !k2
         enddo !k1

         do k1 = 1, m
              do k2 = k1+1, m
              H(k1,k2) = H(k2,k1)
              enddo !k2
         enddo !k1

         time_7 = etime(ustime)
         time_17 = ustime(1)
         write(*,*) 'other hessian calculations cpu = ', time_7 - time_6
         write(*,*) 'other cpu = ', time_17 - time_16

         cpu_g_h_non_exp = cpu_g_h_non_exp + time_7 - time_6

         return
         end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine inverse(H, Hi, HH, IPOS)

c*****************************************************************************
c     This program inverts the matrix H and returns its inverse.
c*****************************************************************************
```

```
        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision  mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  H(m,m), Hi(m,m), HH(m,2*m)

c
c     Local variables
c
        Double precision ep
        Integer          i, j, k, IPO
c***************************************************************************
c     The program instructions begin here.
c***************************************************************************

      IPOS = 1
      ep = 1.0d-20

      write(*,*) 'inversion epsilon = ', ep
      do 10 i = 1, m
            do 10 j = 1, m
                  HH(i,j) = H(i,j)
10                HH(i,j+m) = 0.0d+0
      do 20 i = 1, m
20          HH(i,i+m) = 1.0d+0
      do 80 i = 1, m
      if (HH(i,i) .gt. ep) go to 40
      if (HH(i,i) .lt. -ep) go to 30
      IPOS = 0
      return
30    IPOS = -1
40    do 50 k = i+1, m+i
50           HH(i,k) = HH(i,k) / HH(i,i)
      do 60 j=1,m
         if (j .eq. i) go to 60

      do 55 k = i+1, m+i
55          HH(j,k) = HH(j,k) - HH(i,k)*HH(j,i)
60    continue
80    continue
      do 100 i = 1, m
      do 100 j = 1, m
100          Hi(i,j) = HH(i,j+m)
      return
      end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine interval(x, c, a, b, d, z, v, current_value, tmin,
     1                      value_tmin, tmax, value_tmax, iter)

c***************************************************************************
c     This program control the CS subroutines
c     For ease of handlin the overflow problem, tmin is set at 0.0d+0.
c***************************************************************************

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  x(m), c(n), b(m)
        Double precision  d(m), z(m)
        Double precision  a(m,n)
        Double precision  v(m)
        Double precision  tmin, tmax
        Double precision  value_tmin, value_tmax, current_value
        Double precision  t_small, t_huge, value_t_small

c     Local variables

        Double precision step, previous_step, previous_value
        Double precision value, value1, new_value, value3
        Integer          i, j, k
        Integer          iter
        Integer          num_expansions, num_shrinkages
        Integer          i_overflow

c*********************************************************************
c     The instructions begin here.
c     Note that pass only the working array to "objective".
c*********************************************************************

c   For now, tmin is set at 0 and value_tmin is simply current_value
c   Pass these values back to le_control.

         tmin = 0.0d+0
         value_tmin = current_value

         tmax = 0.0d+0
         value_tmax = 0.0d+0

         previous_step = 0.0d+0
         num_expansions = 0
         num_shrinkages = 0
         step   = 1.0 / (m * iter)
c        step   = 1000.0

         value  = 0.0d+0
```

```
          value1 = current_value
          new_value = 0.0d+0
          value3 = 0.0d+0

c         do i =1, m
c              v(i) = x(i)
c         enddo !i

c         call objective( v, c, a, b, value)

c         write(*,*) 'the current objective value =      ', value1

          do i = 1, m
               v(i) = x(i) + step * d(i) + 0.5 * step * step * z(i)
          enddo !i
          i_overflow = 0
          call objective_test( v, c, a, b, value, i_overflow)

          if (i_overflow .eq. 0) then
c            write(*,*) 'No overflow!'
c     Now, if the value at the step is larger than the current_value,
c     then is search is complete.  Otherwise, need to expand the step
c     so as to find a point beyond the step at which the objective does not
c     suffer from the overflow problem and is larger the objective at the
c     step (not the current_value).
             go to 200
          else
             write(*,*) 'POSSIBLE OVERFLOW DETECTED!'
c     Now, find a point in between 0 and step at which
c     the objective does not suffer from overflow problem and
c     is larger than the current_value.
             go to 300
          endif

200       new_value = value
          if (new_value .gt. value1)  then
               tmax = step
               value_tmax = new_value
               write(*,*) 'tmax is found'
               write(*,*) 'tmin, tmax = ', tmin, tmax
               go to 1000
          endif

250       previous_step = step
          previous_value = new_value
          step = step * 2

          do i = 1, m
               v(i) = x(i) + step * d(i) + 0.5 * step * step * z(i)
          enddo !i
          i_overflow = 0
          call objective_test( v, c, a, b, value, i_overflow)

          if (i_overflow .eq. 0) then
c            write(*,*) 'No overflow!'
             new_value = value
             if (new_value .gt. previous_value)  then
```

```
                tmax = step
                value_tmax = new_value
                write(*,*) 'tmax is found'
                write(*,*) 'tmin, tmax = ', tmin, tmax
                go to 1000
            else
                go to 250
            endif
         else
            t_small = previous_step
            value_t_small = previous_value
            t_huge = step
            call overflow( x, c, a, b, d, z, v, t_small,
     1                    value_t_small, t_huge, tmax, value_tmax)
            go to 1000
         endif

300      t_small = tmin
         value_t_small = current_value
         t_huge = step
         call overflow( x, c, a, b, d, z, v, t_small,
     1                    value_t_small, t_huge, tmax, value_tmax)
         go to 1000


1000     continue
         return
         end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine objective_test(v, c, a, b, value, i_overflow)

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  c(n), b(m)
        Double precision  a(m,n)
        Double precision  v(m)

c     Local variables

        Double precision value, w
        Integer          i, j, k
        Integer          i_overflow

c******************************************************************
```

```
c     The instructions begin here.
c     Note that pass only the working array to "objective".
c*******************************************************************

      i_overflow = 0
      value = 0.0d+0
      do j = 1, n
           w = 0.0d+0
           do i = 1, m
                w = w + a(i,j) * v(i)
           enddo !i
           w = ((w - c(j)) / mu ) - 1.0

c      return if w exceeds 20 for possible overflow

           if (w .gt. 20.0) then
c             write(*,*) 'Large Exponent (>20)!  exponent = ', w
              i_overflow = 1
              go to 1000
           endif
           value = value + exp (w)
       enddo !j

       w = 0.0d+0
       do i = 1, m
            w = w + b(i) * v(i)
       enddo !i

       value = -1.0 * (w - mu * value)

1000   continue

       return
       end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine overflow( x, c, a, b, d, z, v, t_small,
     1                      value_t_small, t_huge, tmax, value_tmax)

c**************************************************************************
c       This program finds a point, tmax, between t_small and t_huge
c       at which (1) the objective value does not suffer from
c                    possoble overflow;
c                (2) the objective value is greater than value_t_small.
c                    value_t_small is updated iteratively in the program
c                    because all we need is a three-point pattern.
c*************************************************************************

        Implicit none

        integer m,n
```

```
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  x(m), c(n), b(m)
        Double precision  d(m), z(m)
        Double precision  a(m,n)
        Double precision  v(m)
        Double precision  tmin, tmax
        Double precision  value_tmax
        Double precision  t_small, t_huge
        Double precision  value_t_small

c     Local variables

        Double precision step, previous_step
        Double precision value, value1, new_value, value3
        Integer          i, iter
        Integer          i_overflow

c**********************************************************************
c     The instructions begin here.
c     Note that pass only the working array to "objective".
c**********************************************************************


d       write(*,*) 'entering overlow'
1        step = 0.5 * (t_small + t_huge)
         do i = 1, m
              v(i) = x(i) + step * d(i) + 0.5 * step * step * z(i)
         enddo !i

         i_overflow = 0
d    write(*,*) 'before the objective_test in overflow'
         call objective_test( v, c, a, b, value, i_overflow)
d    write(*,*) 'before the objective_test in overflow'
         if(i_overflow .eq. 0) then
            if (value .gt. value_t_small) then
               tmax = step
               value_tmax = value
               go to 1000
            else
               t_small = step
               value_t_small = value
               go to 1
            endif
         else
            t_huge = step
            go to 1
         endif

1000     continue
         return
         end
```

PROPRAM DELIMITER - PROPRAM DELIMITER - PROPRAM DELIMITER

```
      subroutine golden( x, c, a, b, d, z, v, tmin, value_tmin,
     1                    tmax, value_tmax, t, value_t, epsilon_l)

c***************************************************************************
c     This program performs golden-section search.
c     The output is t and value_t, the optimal step in the current
c     direction and the objective value at t.
c     Note: the output tmin and tmax values are different from the
c     input ones.
c***************************************************************************

        Implicit none

        integer m,n
        common /sizes/ m,n

        double precision mu
        common /perturbation/ mu

c     The following arrays are allocated by the c_control.c program

        Double precision  x(m), c(n), b(m)
        Double precision  a(m,n)
        Double precision  d(m), z(m)
        Double precision  v(m)

c     Local variables

        Double precision gold
        Double precision uncertainty
        Double precision value_t, value_tmin, value_tmax
        Double precision value, valuea, value1, value2, valueb
        Double precision tmin, tmax, t1, t2, t, epsilon_l

        Integer          i, j, k

c*******************************************************************
c     The instructions begin here.
c     Note that pass only the working array to "objective".
c*******************************************************************

d       write(*,*) 'golden-section search begins'

        k = 0

        gold = 0.6180339

d       write(*,*) 'tmin, tmax = ', tmin, tmax
        uncertainty = tmax - tmin

        t1 = tmax - gold * uncertainty
```

```
        t2 = tmin + gold * uncertainty

c       do i = 1, m
c            v(i) = x(i) + tmin * d(i) + 0.5 * tmin * tmin * z(i)
c       enddo !i

c       call objective(v, c, a, b, value)
        valuea = value_tmin

c       call objective(v, c, a, b, value)
        valueb = value_tmax

        do i = 1, m
             v(i) = x(i) + t1 * d(i) + 0.5 * t1 * t1 * z(i)
        enddo !i

        call objective(v, c, a, b, value)
        value1 = value

        do i = 1, m
             v(i) = x(i) + t2 * d(i) + 0.5 * t2 * t2 * z(i)
        enddo !i

        call objective(v, c, a, b, value)
        value2 = value

        k = k + 1

100     If (abs(value1 - value2) .lt. epsilon_l) then
           if ((value1 .le. valuea) .and. (value1 .le. value1) .and.
     1         (value1 .le. value2) .and. (value1 .le. valueb)) then
               t = t1
               value_t = value1
               go to 200
           endif
           if ((value2 .le. valuea) .and. (value2 .le. value1) .and.
     1         (value2 .le. value2) .and. (value2 .le. valueb)) then
               t = t2
               value_t = value2
               go to 200
           endif
           if ((valueb .le. valuea) .and. (valueb .le. value1) .and.
     1         (valueb .le. value2) .and. (valueb .le. valueb)) then
               t = tmax
               value_t = valueb
               go to 200
           endif
           if ((valuea .le. valuea) .and. (valuea .le. value1) .and.
     1         (valuea .le. value2) .and. (valuea .le. valueb)) then
               t = tmin
               value_t = valuea
               go to 200
           endif

        else
           go to 300
        endif
```

```
200     value_tmin = valuea
        value_tmax = valueb
        go to 1000

300     continue

        If (value1 .lt. value2) then
           tmin = tmin
           valuea = valuea
           tmax = t2
           valueb = value2
           uncertainty = tmax - tmin
           t2   = t1
           value2 = value1
           t1   = tmax - gold * uncertainty
           do i = 1, m
                v(i) = x(i) + t1 * d(i) + 0.5 * t1 * t1 * z(i)
           enddo !i
d          write(*,*) 'golden - toward the origin'
           call objective(v, c, a, b, value)
d          write(*,*) 'objective value = ', value
           value1 = value
           k = k + 1
           go to 100
        else
           tmax = tmax
           valueb = valueb
           tmin = t1
           valuea = value1
           uncertainty = tmax - tmin
           t1   = t2
           value1 = value2
           t2   = tmin + gold * uncertainty
           do i = 1, m
                v(i) = x(i) + t2 * d(i) + 0.5 * t2 * t2 * z(i)
           enddo !i
d          write(*,*) 'golden - away from the origin'
           call objective(v, c, a, b, value)
d          write(*,*) 'objective value = ', value
           value2 = value
           k = k + 1
           go to 100
         endif

1000     write (*,*) 'Golden-Section Search concluded.'
         write (*,*) 'Number of interval reductions = ', k, ' .'
         write (*,*) 'Uncertainty = ', uncertainty
d        write (*,*) 'Value at tmin (i.e. valuea)= ', valuea
d        write (*,*) 'Value at t1   (i.e. value1)= ', value1
d        write (*,*) 'Value at t2   (i.e. value2)= ', value2
d        write (*,*) 'Value at tmax (i.e. valueb)= ', valueb
         return
         end
```

Appendix C.4: Computer Programs for Performance Evaluation and Comparison

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define MAXLEN 100      /* max size of number */

double matrix[MAXLEN][MAXLEN];
double distance[MAXLEN], time_sta[MAXLEN], temp_time, total_time,
abus_equipment_time, rail_equipment_time, sta_pass[MAXLEN], scale_factor;

main(int argc, char *argv[]){
        FILE *ifp, *ofp;
        int i, j, k, bus_enter[MAXLEN], bus_length[MAXLEN], unit_fuel_cost;
        int station_num, running_time, headway, shuttle_times, num_partial_route,
enter_point, bus_capacity;
        char c, s[MAXLEN];
        double temp_num, real_num, debug, speed, hour_capacity, customer_demand,
total_fuel_cost, fuel_saving, total_before_round, total_after_round;



        for(i=0; i<MAXLEN; i++){
                bus_enter[i] = 0;
                sta_pass[i] = 0;
        }

        ifp = fopen(argv[1], "r");
        ofp = fopen(argv[2], "w");

        c = getc(ifp);
        while(c != EOF){
                // skip the spaces or TAB or the new line char,
                //until next number read
                while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                        c=fgetc(ifp);
                s[1] = '\0';

                i = 0;
                /* get the number of stations */
                if(isdigit(c))    //collect the integer digits of each number
                                  //put them in an array
                        while(isdigit(s[++i]=c=getc(ifp)))
                        ;
                s[i] = '\0';
```

```
                station_num = atoi(s); //convert the array into the integer value


    while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                        c=fgetc(ifp);
                s[1] = '\0';

                i = 0;
                /* get the scale-up factor */
                if(isdigit(c))    //collect the integer digits of each number
                                  //put them in an array
                        while(isdigit(s[++i]=c=getc(ifp)))
                        ;
                if(c=='.')//collect the fractional part numbers
                        while(isdigit(s[++i]=c=getc(ifp)))
                                  ;
                s[i] = '\0';
                scale_factor = atof(s); //convert the array into the integer value


                for(k=0;k<(station_num-1);k++){
                        while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                        c=fgetc(ifp);
                        s[1] = '\0';

                        i = 0;
                        /* get the distance between each station */
                        if(isdigit(c))    //collect the integer digits of each number
                                //put them in an array
                                while(isdigit(s[++i]=c=getc(ifp)))
                                ;
                        if(c=='.')//collect the fractional part numbers
                                while(isdigit(s[++i]=c=getc(ifp)))
                                ;
                        s[i] = '\0';
                        distance[k] = atof(s);   //convert the array into the real number
value
                }

                while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                        c=fgetc(ifp);
                s[1] = '\0';

                i = 0;
                /* get the total running time */
                if(isdigit(c))    //collect the integer digits of each number
```

```
                        //put them in an array
                while(isdigit(s[++i]=c=getc(ifp)))
                ;
        s[i] = '\0';
        running_time = atoi(s);         //convert the array into the integer value


        while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                c=fgetc(ifp);
        s[1] = '\0';

        i = 0;
        /* get the headway */
        if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                while(isdigit(s[++i]=c=getc(ifp)))
                ;
        s[i] = '\0';
        headway = atoi(s);      //convert the array into the integer value


        while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                c=fgetc(ifp);
        s[1] = '\0';

        i = 0;
        /* get the speed */
        if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                while(isdigit(s[++i]=c=getc(ifp)))
                ;
        if(c=='.')//collect the fractional part numbers
                while(isdigit(s[++i]=c=getc(ifp)))
                        ;
        s[i] = '\0';
        speed = atof(s);        //convert the array into the double value


        while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                c=fgetc(ifp);
        s[1] = '\0';

        i = 0;
        /* get the number of partial routes */
        if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
```

```
			while(isdigit(s[++i]=c=getc(ifp)))
			;
		s[i] = '\0';
		num_partial_route = atoi(s);   //convert the array into the integer
value


		j=-1;
		for(k=0;k<num_partial_route*2;k++){
			while((s[0]=c) == ' ' || c == '\t' || c == '\n')
			c=fgetc(ifp);
			s[1] = '\0';

			i = 0;
			/* get the enter/leave point of each partial route */
			if(isdigit(c))    //collect the integer digits of each number
				//put them in an array
				while(isdigit(s[++i]=c=getc(ifp)))
				;
			s[i] = '\0';
			enter_point = atoi(s);  //convert the array into the real number
value
					j = j*(-1);
					bus_enter[enter_point-1] = bus_enter[enter_point-
1]+j;
		}

		bus_length[0] = 1+bus_enter[0];
		for(i=1; i<(station_num-1); i++) {
			bus_length[i] = bus_length[i-1] + bus_enter[i];
		}



		/* read in the matrix */
		total_before_round = 0;
		total_after_round = 0;
		for(i=0; i<(station_num-1); i++){

					for(j=(i+1); j<station_num; j++){
				// skip the spaces or TAB or the new line char,
				//until next number read
				while((s[0]=c) == ' ' || c == '\t' || c == '\n')
					c=fgetc(ifp);
				s[1] = '\0';
```

```
                        k = 0;
                        /* get the numbers */
                        if(isdigit(c))    //collect the integer digits of each number
                                          //put them in an array
                                while(isdigit(s[++k]=c=getc(ifp)))
                                ;
                        if(c=='.')//collect the fractional part numbers
                                while(isdigit(s[++k]=c=getc(ifp)))
                                ;

                        s[k]= '\0';
                        real_num = atof(s)/(running_time/60)*scale_factor;
                        total_before_round = total_before_round + real_num;
                        temp_num = floor(real_num);
                        if((real_num-temp_num)>0.40)
                            matrix[i][j] = temp_num + 1;
                        else
                            matrix[i][j] = temp_num;
                            total_after_round = total_after_round + matrix[i][j];
                            }
                        }



                if((c=getc(ifp)) == EOF)
                    break;

            }//end of while(c!=EOF)



            for (i=0; i<(station_num-1); i++){
                time_sta[i]=distance[i]/speed;
            }

            total_time = 0;
            for (i=0; i<(station_num-1); i++){
                temp_time=0;
                for (j=(i+1); j<station_num; j++){
                    temp_time=temp_time+time_sta[j-1];
                    total_time=total_time+temp_time*matrix[i][j];
                }
            }


    /* capacity check */
```

```
	bus_capacity = 50;
	hour_capacity = (60/headway)*bus_capacity;
	for(i=0; i<(station_num-1); i++){
		customer_demand = 0;
		for(j=0; j<=i; j++){
			for(k=(i+1); k<station_num; k++){
				customer_demand = customer_demand + matrix[j][k];
			}
		}
		sta_pass[i] = customer_demand;
		if(customer_demand>(hour_capacity*bus_length[i])){
			fprintf(ofp, "Overflow at station\t%d\n", i+1);
			exit(0);
		}
	}


  /* fuel saving */
  fuel_saving = 0;
  total_fuel_cost = 0;
  unit_fuel_cost = 10;
  for(i=0; i<(station_num-1); i++){
	total_fuel_cost = total_fuel_cost + 
distance[i]*unit_fuel_cost*bus_length[i]*(60/headway);
	fuel_saving = fuel_saving + 
(bus_length[i]-1)*0.1*unit_fuel_cost*distance[i]*(60/headway);
  }


  /* ABUS equipment time */
  for(i=0; i<(station_num-1); i++){
	abus_equipment_time = abus_equipment_time + 
(distance[i]/speed)*bus_length[i]*(60/headway);
	rail_equipment_time = rail_equipment_time + 
(distance[i]/speed)*4*(60/headway);
  }


	/* print out the number of stations */
	fprintf(ofp, "Number of stations\t%d\n", station_num);

	/* print out the running time and headway */
	fprintf(ofp, "Running time, speed and headway\t%d\t%d\t%4.1f\n", 
running_time, headway, speed);

	/* print out the distance bwtween 2 stations */
```

```
	fprintf(ofp, "Distance between each station\t");
	for(i=0;i<(station_num-1);i++){
		fprintf(ofp, "\t%4.1f", distance[i]);
	}
	fprintf(ofp, "\n");

	fprintf(ofp, "Travelling time between each station\t\t\t");
	for(i=0;i<(station_num-1);i++){
		fprintf(ofp, "\t%4.1f", time_sta[i]);
	}
	fprintf(ofp, "\n");


	fprintf(ofp, "Bus length between each station\t\t\t");
	for(i=0; i<(station_num-1); i++){
		fprintf(ofp, "\t%d", bus_length[i]);
	}
	fprintf(ofp, "\n");


	/* print out the matrix */
	fprintf(ofp, "The OD Matrix\n");
	for(i=0; i<station_num; i++){
		for(j=0; j<station_num; j++)
			fprintf(ofp, "\t%4.1f", matrix[i][j]);
		fprintf(ofp, "\n");
	}
	fprintf(ofp, "Hourly OD demand and Rounded hourly OD
demand\t\t\t%4.1f\t\t%4.1f\n", total_before_round, total_after_round);
	fprintf(ofp, "Hourly total customer demand at each station\t");
	for(i=0;i<(station_num-1);i++){
		fprintf(ofp, "\t%4.1f", sta_pass[i]);
	}
	fprintf(ofp, "\n");

	/* print out the shuttle's revenue time */
	shuttle_times = running_time/headway;
	fprintf(ofp, "\n");
	fprintf(ofp, "The shuttle's running time is\t%d\n", running_time);
	fprintf(ofp, "\n");
	fprintf(ofp, "The hourly total passenger's travelling time is\t%4.1f\n",
total_time);
	fprintf(ofp, "\n");
	fprintf(ofp, "\t\t\t\t\t\t\t\t\t\t\t\tLight Rail\t\t\tClustered
Bus\t\t\t\tABUS\n");
	fprintf(ofp, "Hourly Total Equipment
```

```
Time\t\t%4.1f\t\t\t\t\t\t%4.1f\t\t\t\t\t\t\t%4.1f\n", rail_equipment_time,
abus_equipment_time, abus_equipment_time);
        fprintf(ofp, "\n");
        fprintf(ofp, "Hourly Total Labor
Time\t\t\t\t%4.1f\t\t\t\t\t\t%4.1f\t\t\t\t\t\t\t%4.1f\n",
rail_equipment_time/4, abus_equipment_time, rail_equipment_time/4);
        fprintf(ofp, "\n");
        fprintf(ofp, "Hourly Total Fuel
Cost\t\t\t\t\t\t\t\t\t\t\t\t%4.1f\t\t\t\t\t\t\t%4.1f\n", total_fuel_cost,
(total_fuel_cost-fuel_saving));
        fprintf(ofp, "\n");
        return 0;
}
```

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

PART III - EVALUATION OF URBAN BUS AUTOMATION: DISTURBANCE TO SURROUNDING TRAFFIC

EVALUATION OF BUS AUTOMATION SCENARIOS:
MINIMIZATION OF DISTURBANCE TO SURROUNDING TRAFFIC DUE TO TRANSIT SIGNALING PRIORITY THROUGH BUS AUTOMATION

Lin Lin and H.-S. Jacob Tsao
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 95192-0085

TABLE OF CONTENTS

PART III - EVALUATION OF URBAN BUS AUTOMATION: DISTURBANCE TO SURROUNDING TRAFFIC

EVALUATION OF BUS AUTOMATION SCENARIOS: MINIMIZATION OF DISTURBANCE TO SURROUNDING TRAFFIC DUE TO TRANSIT SIGNALING PRIORITY THROUGH BUS AUTOMATION

Lin Lin and H.-S. Jacob Tsao
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 95192-0085

Abstract: Light-rail operations suffer from the major problem of physical disconnection from local collection and distribution of passengers. Replacing light-rail cars with buses can solve part of the problem. Automated lateral control can reduce the lane width required so that the light-rail right-of-way suffices. Automated closely-spaced bus convoying can mimic the train operation of light-rail. A key performance measure is the disturbance to the surrounding traffic caused by transit signal priority (TSP). This paper shows that automation significantly reduces TSP disturbance caused by bus operations.

1. INTRODUCTION

Traffic congestion will continue to worsen and likely worsen at a faster rate than ever. With the completion of the construction of the National Highway System and the general lack of available right-of-way for adding lanes on existing freeways in the largest metropolitan areas around the nation, the issue of traffic congestion has received more and more attention. A promising approach to solving this problem of traffic congestion is to use or develop advanced technologies to improve the capacity of the current transportation systems or the efficiency of using the current capacity. This approach has led to a large group of operating concepts and user services called Intelligent Transportation Systems (ITS). Many efforts have focused on increasing vehicle throughput, particularly automobile throughput. The most forward-looking and technology-intensive ITS is the concept of automated driving.

Automated driving has been treated primarily as a means to increase automobile throughput on the nation's highways, and that line of research and development has been referred to as Automated Highway Systems (AHS). The high potential for increasing automobile throughput is accompanied by a high level of risk resulting from the complexity of the technical, institutional and political issues involved in the design of a deployable system and in its staged deployment (Tsao et al., 1994; Tsao and Ran, 1996; Chen and Litkouhi, 1998; Tsao, 1998a; Tsao, 2001). Realizing the complexity of the issues, the ITS research community has reduced its attention to AHS significantly. For example, the support by the federal government for seven-year program of the National Automated Highway System Consortium was withdrawn after two years of research development, due to complexity, organization and other issues.

This paper focuses on the potential of automated driving on urban transit operations on city streets. The authors believe that automated transit operations on urban streets could be a more promising concept, not only as an "end-state" by itself but also as an intermediate step toward the implementation of an AHS accommodating also automobiles. The authors proposed a transit service for AHS debut in a paper published in the IVHS Journal in 1995 (Tsao, 1995d). As a firm believer of incremental deployment of vehicle-automation technologies, the authors argued for a balanced approach between "market pull" and "technology push," e.g., (Tsao, 1995b), (Tsao, 1998a), (Tsao, 1998b), (Tsao, 2001), and (Tsao and Botha, 2001). Tsao and Botha (2003) proposed operating concepts for urban bus automation as well as deployment steps toward them.

1.1 Motivation of an Automated BUs System (ABUS)

Research and implementation efforts on bus automation began at least a quarter century ago under the umbrella of dual-mode (bus) transportation. See, for example, (DeMarco, 1974). More recent efforts include the implementation of a Bus Rapid Transit (BRT) system in Adelaide, Australia, (South Australia DOT, 1988) and the study of a guided bus system in Eugene/Springfield area of Oregon (Carey et al., 1998).

Several operating concepts for Automated BUs Systems (ABUS) have been proposed in (Tsao and Botha,2003). An ABUS is any bus system that supports hands-off or feet-off driving. These concepts were motivated to combine the strengths of bus and light-rail operations by mimicking the light-rail operations with buses on the line-haul section and by using the same buses for local collection and distribution of passengers. Where the available right-of-way in the median of a city boulevard is too narrow for manually-driven bus operations, a busway enabled by precision lane-keeping, precision turning and precision docking, which is in turn enabled by automated lateral vehicle control, may be a viable option. Automated closely-spaced bus convoying enabled by electronic coupling, which is in turned enabled by bus-to-bus communication and automated longitudinal vehicle control, can mimic the train operations of light-rail. This type of virtual-train operations has the potential of allowing driverless operations for the trailing buses of an automated bus convoy and hence the potential of rivaling the light-rail operations in the labor cost. (A light-rail train is operated by one driver, and part of the reason why this can be safely achieved is that the light-rail cars are articulated and the absence of inter-car distance enables the driver to supervise the train operation safely.)

Light-rail operations often enjoy transit signal priority or even preemption, which does disturb the surrounding traffic. Automated closely-spaced bus convoying has the potential of not exceeding the degree of disturbance to the surrounding traffic achieved by the light-rail operations. While transit signal priority is justified for achieving faster travel time of higher-occupancy transit vehicles, disturbance to surrounding traffic caused by such priority or preemption has been a major concern of automobile drivers and traffic engineers. This important issue has received some attention in the literature, e.g., Jacobson and Sheffi (1981) and Sunkari et al. (1995). However, results reported in the literature are mostly analytical models. The stochastic nature of the disturbance to surrounding traffic makes the estimation problem particularly complex, especially for the approach of analytical modeling. The existing models have yet to be verified empirically or via computer simulation. Such disturbance to surrounding traffic is the main focus of this paper. We built simulation models and used computer simulation to obtain detailed results.

Although such an ABUS system has these high potentials of mimicking the light-rail operations along the line-haul section (and the high potential of using the same buses for local pick-up and distribution), implementing the system may require incremental steps. Tsao and Botha (2003) developed a four-step deployment sequence leading to such an ABUS system. This paper focuses on the improvement in disturbance to surrounding traffic as the deployment progress from a conventional busway dedicated to end-to-end conventional shuttle buses through the sequence toward an ABUS.

Automated longitudinal control enables better and perhaps seamless coordination between bus movement and traffic signaling, which improves travel time and may achieve a lower degree of disturbance to the surrounding traffic than the conventional light-rail operations. Studying this potential requires detailed synchronization algorithm, and hence is beyond the scope of this paper.

1.2 Motivation and Contribution of This Study

This report presents a simulation study of the traffic disturbance at intersections along an urban commute thoroughfare. The study compares the ABUS operation with the three other scenarios, including a base condition and two intermediate steps toward the ABUS with respect to one performance measure - the disturbance of the signal preemption to the surrounding traffic. Their comparisons with respect to other performance measures, e.g., equipment requirement, labor requirement and fuel efficiency, are reported separately. Those comparisons can be made using deterministic approaches while the disturbance to surrounding traffic is by nature a stochastic phenomenon and an accurate comparison requires computer simulation.

The expected amount of time a motor vehicle spends to pass the intersection is the statistic collected to measure the performance of the ABUS and its intermediate steps.

Abstracted from real world practices and observations, a number of hypothetical scenarios of traffic volume and traffic signaling cycles are discussed and assumed in this paper to simulate the traffic conditions at the urban intersection.

This study also shows the feasibility of event-driven simulation for studying the length of the time a vehicle spends in passing an intersection. Discrete-even systems simulation is the modeling of the systems in which the state of the system changes only at a discrete set of points in time. In other word, time is advanced from event to event instead of continuously or in short and constant increments. The Promodel simulation model is used as a tool to build simulation models for the study. Discrete-event systems simulation has been successfully used to model and study vehicle traffic, e.g., Hall et al. 2001 and Hall et al. 1997.

As the study shows, the disturbance to the surrounding traffic caused by signal priority decreases as the system progresses from the conventional busway toward the ABUS.

1.3 Objective of the Study

The objective of this study is to use computer simulation to conduct a quantitative study in order to determine if the deployment of the proposed ABUS operation concepts through the steps would decrease the disturbance to the surrounding traffic, in particular to the cross traffic, and, if so, estimate the amount of reduction to the disturbance associated with each intermediate step.

2. PROBLEM STATEMENET AND FORMULATION

2.1 A Four-step Deployment Sequence of the ABUS system

Believing that vehicle automation has a high potential for improving the efficiency of urban transit operations, Tsao and Botha (2001) developed the concept of ABUS as well as four intermediate deployment steps. The four steps and their main features are summarized in Table 1.

Note the both Step 3 and Step 4 are a ABUS system because both involve (automated) hands-off and feet-off bus driving. The only difference between Step 3 and Step 4 is that in Step 4 no drivers are needed in the trailing buses of a bus convoy. Since the focus of this paper is on the disturbance to surrounding traffic and Step 3 and Step 4 are no different in this aspect. We will compare the ABUS (of either Step 3 or Step 4) to a base case where is no transit signal priority/preemption and to the first two intermediate steps to the ABUS.

2.2 Geometry of the ABUS Operating Concepts

We focus on a commute thoroughfare rather than a large-scale metropolitan-wide bus system. The commute thoroughfare may comprise of as many numbers of intersections as desired.

Figure 1 presents the geometry of the intersections simulated. The intersection is formed by two streets with four entry/exit sides marked as A, B, C and D. These marks are used to indicate traffic directions. For example, the vector AC represents the through traffic moving from the A side to the C side, and the vector CB represents the left-turning traffic from side C to side B.

Each side of the intersection is provided one left lane, one right lane and one go-straight lane, for each traffic direction. Each lane has a capacity of holding 5 vehicles (approximate 50 feet length-wise) at most when there exists a queue. All conventional vehicles, also referred to as non-Abus vehicles, will enter either of three lanes first before passing the intersection. These lanes serve as a start point for vehicles to enter the intersection, shown as “Start” in Figure 1. Lanes for vehicles coming from the other three sides to leave the current intersection are grouped and shown as “Receive” in Figure 1.

In the median of the entire length of road along the thoroughfare, an exclusive bus lane is built, one for each of the two directions. In an ABUS system, this bus lane is used exclusively by the automated bus convoys. In any of the intermediate steps, it is used as a conventional bus way. Figure 1 represents the road configuration of an ABUS system, and hence this bus lane is labeled as the ABUS lane.

2.3 Signal Priority-Preemption for Bus Operations of ABUS and All Steps

Buses traveling on the exclusive bus lane, in the ABUS system or any of the three intermediate steps, is given full priority over all conventional vehicles, in passing the intersection. Therefore, two sets of signaling programs are implemented at each interaction to control the traffic lights. One is called the Regular program while the other is called the Advanced program. Signals under the Regular Program are referred to as Regular signals, and those under the Advanced Program are referred to as Advanced signals.

The Regular Program is on when there is no bus approaching the intersection in the bus lane, while the Advanced Program is activated whenever a bus approaches the intersection. Under the Regular Program, the traffic lights change color between Green, Yellow and Red in a fixed cycle. The change of each and every color is running cyclically at a fixed frequency, and around the clock time constantly. The duration of each color is also fixed marking a fixed cycle time of color changes.

Whenever the system has sensed the approach of a bus on the bus lane, the signal preemption is triggered. The preemption is triggered three seconds before the bus acrtually arrives to pass the intersection, for safety consideration. The Advanced Program ensures all signals receive instruction of a designated color simultaneously at the instant when the system has sensed the approach of the bus, no matter what color these signals were displaying just before the approach of the bus. The change of color under this program is non-cyclic, nor at a fixed frequency. The occurrence of the change is not around the clock time, rather, it is dependent of the buses’ arrival times. Each signal will retain in its assigned color for the entire interval starting from the time of preemption triggering till the time that the bus completes its movement through the intersection.

Table 1: A Four-Step Deployment Sequence for an ABUS System

Step	Main Features	Main Benefits	Main Traffic Issues
1 Conventional Busway (with sufficient right-of-way)	• Bus operations on light-rail right-of-way	• Same bus for line-haul and collection/distribution	• Scattered buses disturbing surrounding traffic
1' Busway with Automation to Enable Bus Operations on Narrow Right-of-way	• Automated precision turning/automated precision lane-changing • Automated lane-keeping • Automated bus-following • Automated precision docking (if desired)	• Same bus for line-haul and collection/distribution • Automation enabling busway operations on narrow right-of-way	• Scattered buses disturbing surrounding traffic
2 Manual Bus Convoying Through ITS	• Clustering buses to reduce disturbance via ITS technologies	• Reduction of disturbance to surrounding traffic due to signal preemption or priority	• Possible excessive disturbance as demand grows
3 Automated Closely-spaced Convoying	• Automated closely-spaced convoying • Automated precision turning/precision lane-changing, automated lane-keeping, automated vehicle-following and precision docking if not already implemented	• Reduction of disturbance to surrounding traffic due to signal preemption or priority	• Safety
4 Driverless Bus-following	• Absence of driver on trailing buses	• Reduction of labor cost	• Safety

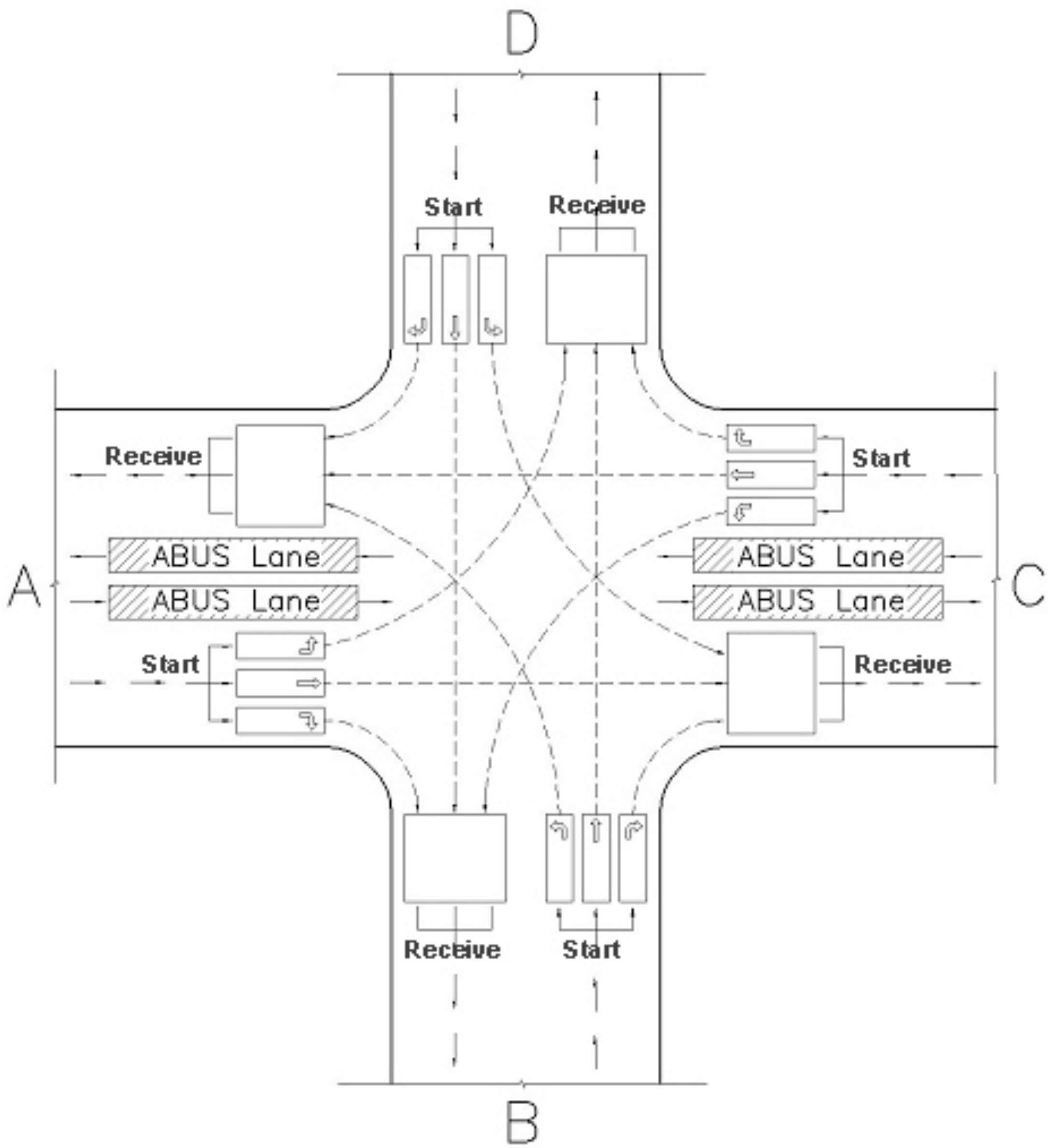


Figure 1: Geometry of the Intersection

2.4 Traffic

2.4.1 Bus and Others

Vehicular traffic is grouped in two categories: buses using the exclusive bus lane and all other vehicles using the other lanes. For ease of discussion, we reserve the term bus for the exclusive reference to a bus using the exclusive bus lane. A bus has its exclusive way and does not wait behind any non-Abus vehicles. At the intersection, like any conventional vehicle, a bus may choose to go straight, or turn left or turn right, to pass the intersection, according to the known turning probabilities, with which all vehicles entering the intersection will go straight through, turn left or turn right, respectively.

As will be discussed next, the turning probabilities are part of the parameters used in this study to describe the traffic volume at an intersection. We assume that both buses and other vehicles follow the same probability distribution of turning.

2.4.2 Natural Platooning and Passing

The natural platooning, i.e., vehicles clustering behind a slowly moving vehicle, is defined to occur only within a small portion of the road that is close to the intersection. This is assumed because overtaking or passing usually occurs farther upstream from the intersection. The situation along the entire length of corridor is beyond the concern of the study.

As stated earlier, the intersection under study has one lane for turning to the left, one for turning to the right, and one for going straight, at each side. Each lane is capable of holding five vehicles length-wise at most. These lanes are the place where non-bus vehicles utilize to form the platoon in passing the intersection. In another words, a non-bus vehicle will enter either of three lanes first before passing the intersection. Non-bus vehicles in the lane will queue up if they meet a red signal, or will pass the intersection if they have a green signal, according to the FIFO (first in first out) rule. The clustering occurs within the lane such that, if the leader in the lane can not exit the intersection due to a red signal or some other reasons, all its followers including those that are not yet in the lane will have to wait behind it, one after another.

The arrival times at an interaction of two buses coming from two opposite directions (e.g., one being from A to C and the other being from C to A) may be close. The one that arrives later has to wait until the other completes the movement of passing. (Note that in discrete even simulation, time is advanced from event to event. The Promodel identifies arrival time as a discrete occurrence, even though two time points are very close.)

2.4.3 Traffic Volume

The following three criteria are used to measure the traffic volume of an intersection:

1) The frequency of the conventional vehicle arrival at an intersection,

2) The probabilities with which a conventional vehicle entering the intersection will turn left, turn right, or go straight through respectively.(Note that in this study the probability distribution is applied to buses as well.)

3) The duration of a green and a red traffic color under the regular signal control system

In accordance with the above parameters, two different sets of traffic volumes are defined below:

Type A Intersection is a Main *Vs* Main intersection, where two crossing streets are both main streets and equally busy, characterized by the following parameters:

- Both streets have equal arrival rates of vehicles, (higher ***Vs*** higher)

- Both streets have the same turning probability distribution. (higher ***Vs*** higher)

- Both streets are assigned equal durations of green and red signals. (longer ***Vs*** longer),

Type B Intersection is a Main *Vs* Secondary intersection, where one of the two streets is a main street while the other is a secondary street in terms of traffic volume. The two streets are characterized with the following parameters:

- One street (specifically referring to the thoroughfare in this study) has a higher arrival rate of vehicles while the cross street has a lower arrival rate. (higher *Vs* lower)

- The higher probability that vehicles would travel straight through the intersection to continue on the thoroughfare (less likely to turn and join the cross traffic) ***Vs*** lower probability that vehicles would stay on the cross street (more likely to turn either right or left to join the thoroughfare traffic). (higher x lower)

- A longer duration of a green for the thoroughfare traffic ***Vs*** a shorter duration of a green for the traffic of the other street. (longer *Vs* shorter); conversely, a shorter duration of a red signal for the thoroughfare traffic ***Vs*** a longer duration of a red for the traffic of the cross street.

Durations of color signals for both type A and type B intersection are tabulated in Table 5.

2.4.4 Traffic Rules

All vehicles entering an intersection are signaled by the traffic lights to make their movements. There is one set of traffic lights at each and every side of the intersection. The color shown on each and every signal conveys the message of traffic rules that all vehicles should obey. A green color conveys the message of Pass, a red means Stop, and a yellow is set up for safety consideration warning the change from a green to a red.

As stated earlier, two signaling programs are programmed to perform the control of vehicle movements through the color change of traffic lights, which results in two different sets of traffic rules.

Traffic rules under the Avanced Program are specified in Table 2, which states that, whenever the signaling system senses the approach of a bus, signal preemption begins, meaning the light that would direct the bus to pass the intersection in a specific direction will turn green, and all the other lights will change their colors accordingly so as to achieve all vehicle movement coordination at the intersection.

Traffic rules under the Regular Program fulfill their functions through either of two modes as described below.

One is named symmetric or type 1 control of traffic rule, in which a green signal will allow for vehicles from two opposite directions of one street to move symmetrically and simultaneously, keeping all vehicles of the other street stop. Each street has one set of lights to control the movement of going straight, and one set of lights to control the movement of turning left. Each green will allow for only one direction of the movement, i.e., either turning left or going-straight, not for both at a time. Two crossing streets take turn to have green signals in a fixed order, so that one cycle of color change completes after two streets have had two sets of greens each.

The other is named asymmetric or type 2 traffic control, in which a green signal will allow for vehicles of one side to pass the intersection, one side at a time keeping all vehicles of the other three sides stopped. Each of four sides on two streets takes turn to have a green so that one cycle of color change completes after all four sides have had a green each. Both left-turn and going-straight movements occur during the same period of green time for each side.

Type 1 and type 2 traffic rules during one signaling cycle are shown in Figure 2 and Figure 3, for both type A (main x main) and type B (main x secondary) intersections. More discussions on traffic rules can be found in section 3.4 and 3.5. Under either type of regular signal control, there is no the presence of any bus and the movements of conventional vehicles turning left or going straight are synchronized with the color of lights. However, no specific rule is imposed to constrain the movement of vehicle turning right when no buses are present or approaching the intersection.

Table 2. Movement Coordination/Relationship between a Bus and the other Vehicles at the Intersection (Traffic Rules under Advanced Signal System, Y=yes =have a green, N=no=have a red)

	Conventional Vehicle's Movement subject to the ABUS movement			
	origin	go through	turn to right	turn to left
if ABUS from A going straight to C	vehicle from A	Y	Y	N
	vehicle from B	N	Y	N
	vehicle from C	Y	Y	N
	vehicle from D	N	Y	N
if ABUS from A turning right to B	vehicle from A	N	N	N
	vehicle from B	N	Y	N
	vehicle from C	Y	Y	N
	vehicle from D	N	Y	N
if ABUS from A turning left to D	vehicle from A	Y	Y	N
	vehicle from B	N	Y	N
	vehicle from C	N	N	N
	vehicle from D	N	Y	N

	origin	go through	turn to right	turn to left
if ABUS from C going straight to A	vehicle from C	Y	Y	N
	vehicle from D	N	Y	N
	vehicle from A	Y	Y	N
	vehicle from B	N	Y	N
if ABUS from C turning right to D	vehicle from C	N	N	N
	vehicle from D	N	Y	N
	vehicle from A	Y	Y	N

	vehicle from B	N	Y	N
if ABUS from C turning left to B	vehicle from C	Y	Y	N
	vehicle from D	N	Y	N
	vehicle from A	N	N	N
	vehicle from B	N	Y	N

2.5 Performance Measures

The performance measure used in this study is the expected amount of time it takes a non-bus vehicle (from either of the A, B, C, D four directions) to pass (i.e., go straight, turn right at or turn left at) the intersection, i.e., the elapsed time between its entry into and exiting from the intersection.

3. MODELING APPROACH

3.1 Physical to Logical Modeling Abstraction

One major abstraction is the fact that all intersections have the same traffic characteristics/properties in nature that can be specified through the aid of mathematical parameters, regardless where they are physically located and/or when they are being studied. Therefore, all intersections are identical unless the parameters used to describe them are changed. This abstraction greatly facilitates the modeling work. As any one of intersections on the corridor is replicable of the others but only the values of the parameters may vary, the object of the study can be thus reduced to a single intersection problem, which obviously mitigates the complexity of dealing with an entire thoroughfare along a long commute corridor that might contain hundreds of intersections. A list of key parameters and their corresponding assumptions describing the nature of the intersections considered in this study are summarized in Table 3.

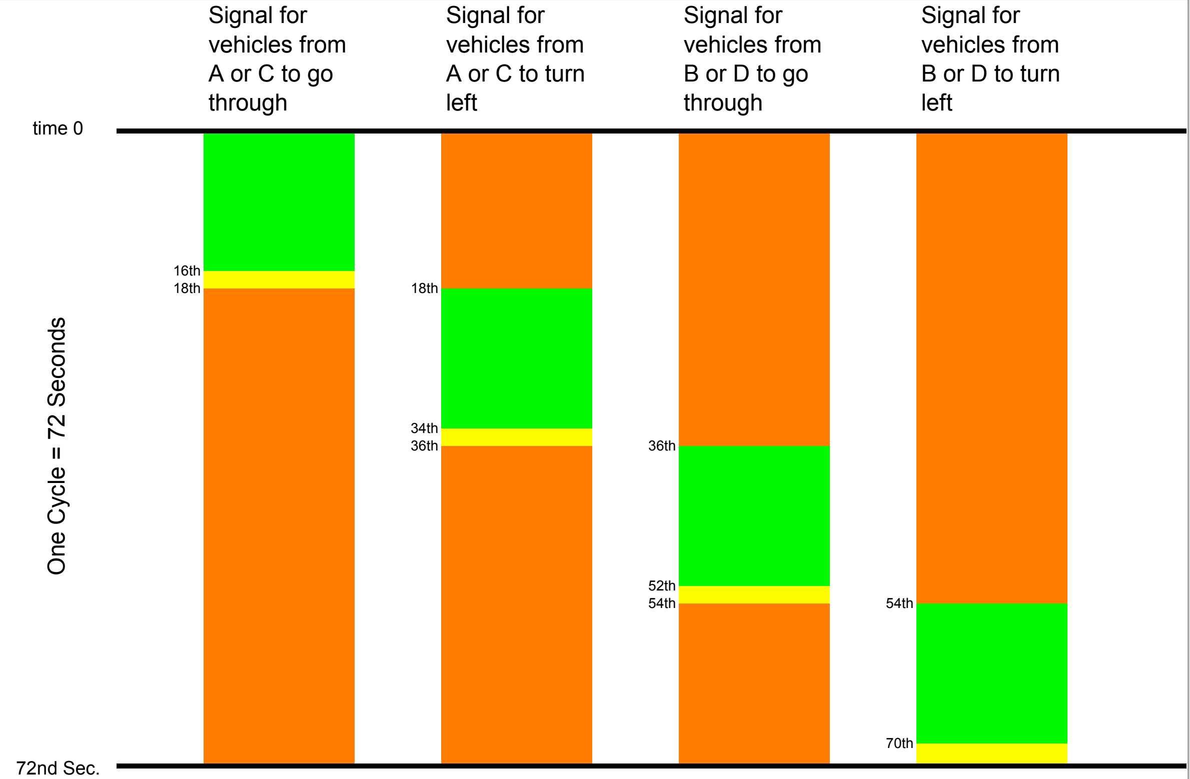


Figure 2: One Cycle Signal Change of Type 1 Traffic Lights (main x main intersection)

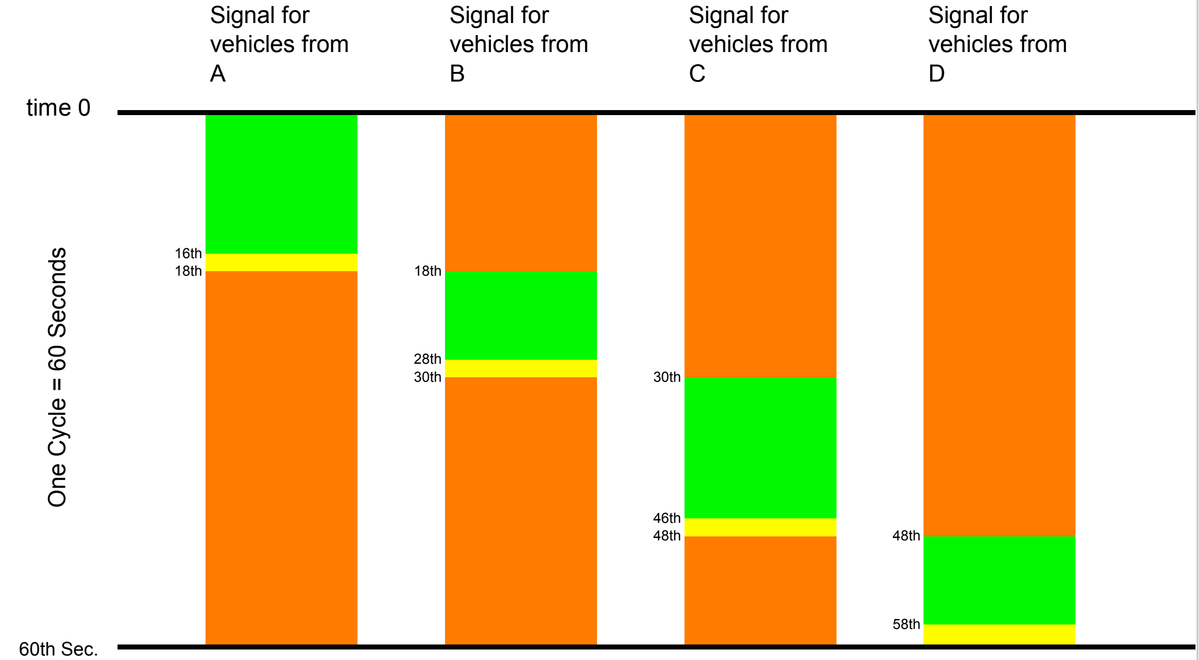


Figure 3: One Cycle Signal Change of Type 2 Traffic Lights (main x secondary intersection)

Another abstraction is the logical interpretation of the operation concepts associated with the four steps toward the ABUS system. Believing that the key factors which make one step distinguishable from another should mainly lie in the arrival pattern of buses at the intersections, it is abstracted that all steps are essentially the same in operations except each has a different bus arrival patterns at the intersections. This arrival pattern is random, and can be characterized by three aspects: the size of platoon or convey (with a single bus being considered as a one-bus platoon and with two or more buses that are intentionally clustered to reduce the disturbance to the surrounding traffic treated and named as a convoy), the type of distribution of the inter-arrival time of platoons or convoys, and the arrival rate of platoons/convoys. The size of platoon/convoy depends on the intermediate step toward the ABUS. The type of the inter-arrival time distribution used in this study is the Normal distribution. The arrival rate of bus platoons/convoys varies. However, for fair comparisons among the different intermediate steps, the assumed demands are identical. For those steps with buses intentionally clustered, the platoon/convoy size is large but the arrival rate of the larger platoons/convoys is small so that the number of buses using the thoroughfare is the same across all four steps.

More explicitly, the difference in the traffic pattern among the four operational steps can be captured by the platoon/convoy size and the Normal inter-arrival time distribution, which is in turn characterized by two parameters: the mean and the variance of the normally distributed random inter-arrival time. Accordingly, each and every step of Abus operation concepts is compared as follows:

> Under step 1 of ABUS operation, the intersection would encounter the highest Abus arrival rate and the greatest variability of this rate of all three steps.
>
> Under step 2 of ABUS operation, the intersection would encounter a lower Abus arrival rate and a lower variability of this rate than that in step one.
>
> Under step 3 of ABUS operation, the intersection would encounter the lowest Abus arrival rate and the least variability of this rate among three steps.
>
> Step 4-Driverless Operation of the ABUS. Because this operation step is mainly concerned with the safety issue and extended on the basis of step 3, therefore, this step will be excluded hereinafter from further study.

In summary, the mean of the bus inter-arrival time increases with respect to the progression of the implementation step. However, the variance of the bus inter-arrival time decreases. The mean and variance of the bus inter-arrival time assumed as model input are summarized in Table 3.

3.2 An Event-Based Simulation Tool – ProModel

Promodel Version 4.2 is used to build the simulation model. Models considered in this study are discrete even-based, in which time is advanced from event to event.

Entities, Locations, Arrivals, Processes, and Resources are the five major components of a simulation study that can be defined with the Promodel Simulation Language, where:

- **Entity** is anything that a model processes.
- **Locations** represent fixed places in the system where entities are routed for processing, storage, or some other activities or decision making.
- Anytime new entities are introduced into the system, it is called an **Arrival.**
- **Processing** defines the routing of entities through the system and the operations that take place at each location they enter. Once the entities have entered the system, as defined in the arrival table, processing specifies everything that happens to them until they exit the system.
- Resource is a person, piece of equipment, or some other device used for functions such as transporting entities and assisting in performing operations entities at locations, and more.

  In this study, all but resources are defined in the model.

3.3 Modeling the Geometry

We focus our discussion on one intersection. An intersection is modeled as a system. Buses and other vehicles are two major entities “being processed” in the system. The process taken place in the system follows simple Enter (vehicle’s arrival), Queue(vehicle’s wait for process), Release(pass the intersection) model.

The pattern of arrivals at the intersection of one entity - the non-bus vehicles - is modeled as a Poisson process. Stated another way, the inter-arrival time of non-bus vehicles at an intersection is assumed to be a random variable that is exponentially distributed. The inter-arrival time of the other entity – buses - at the intersection is modeled as a random variable that follows a Normal distribution. (Note the interarrival time is an attribute of the entity, which can be specified in the arrival table of the Promodel)

The bus lane and all the other lanes (used by non-bus vehicles) are represented by Locations. Different entities have different processes that take place at their corresponding locations throughout the system.

3.4 Modeling of Signaling

As stated in previous sections, two signaling programs (Regular Program and Advanced Program) are implemented at an intersection. The Regular Program produces regular signals that change colors in a cyclic pattern at a fixed frequency, but the Advanced Program produces signals to preempt the regular signals. The first goal of modeling the signaling is to simulate the cyclic color change of the regular signals. The second goal is to simulate the signal preemption.

Under the Regular program, there are two types of traffic rules to perform the control over the vehicle movements. Both types of control require the traffic lights to change color by a certain sequence, repeatedly.

For type 1 signal change rules (or symmetric traffic control rules), each street has one set of lights to allow the non-bus vehicles to go straight through (for two opposite traffic flows) and another set of lights to allow left-turn movements (for two opposite traffic flows). Hence, the intersection has four sets of lights that cyclically alter their colors in sequence. Each set of lights will change color in the order of Green-Yellow-Red, and each color is assigned a certain length of duration. Figure 2 specifies the sequence of the following four sets of signals:

(1) Green color for AC/CA traffic to go straight-

(2) Green color for AC/CA traffic to turn left-

(3) Green color for BD/DB traffic to go straight-

(4) Green for BD/DB traffic to turn left.

Knowing this sequence and knowing that the Promodel is able to define the repetitive occurrence of the vehicle arrivals through defining the entity arrivals, we created four entities with the model, each representing one usage of the above four greens. Furthermore, through defining the frequency and the first time of the occurrence of each and every Green in Promodel's arrival table, the modeling of the cyclic color changes of traffic signals through the clock time is thus achieved. Certainly, entities could be created for the yellow or the red instead of the green but with the same purpose, that is, modeling the cyclic color change of the regular traffic signals.

The modeling of type 2 signal change rules (or asymmetric traffic control rules) is represented in Figure 3. The ideas behind and the procedures for modeling cyclic color change of lights are the same as those for the type 1 signal change rules, except that the sequence of four sets of signals change is changed to:

(1) Green color for vehicles from corner A to go straight or turn left –

(2) Green color for vehicles from corner B to go straight or turn left –

(3) Green color for vehicles from corner C to go straight or turn left –

(4) Green color for vehicles from corner D to go straight or turn left.

Modeling the signal preemption in presence of a bus is achieved by defining the processing/routing for each entity - the green color - through writing the logic/code in the process/routing table of the Promodel. The idea behind these codes is that, after a green color (the entity) arrives at the intersection - the first location that the green enters, it is conditionally given one of two possible routes as follows.

Under the condition that a bus is approaching the intersection (i.e., the system), the entity - the green - will be routed to exit the system immediately without any delay, meaning that the entity will not play any role in the system and no further process is to occur on the entity until next arrival (coming with a pre-specified patter, already defined in the arrival table). This is equivalent to the fact that the function of a green color is made invalid, or that the green is preempted.

Under the other condition, where no bus is approaching the intersection, the entity is routed to next location and is constrained (by logic codes) to stay there for a certain length of time (duration of a green), then routed to exit the system. Such processing/routing reflects the cyclic color change of signals.

3.5 Modeling of Traffic

3.5.1 The Movement of a Single vehicle:

Since vehicles are represented by the entity in the Promodel, modeling the movement of a vehicle is to model the entity's movement by defining the processing for the entity. (Note that only one entity is actually created for all vehicles of the same type.) The movement of a vehicle from one intersection to another is represented in ProModel by the fact the entity is routed from one location to another, as the intersections are represented by locations.

By the same token, the going-straight or turning movements at the intersection can be simulated in the same manner as if the entity is routed from one location to another. For example, the movement of a vehicle from side A turning right to B (in Figure 1) is just modeled as routing the vehicle (the entity) from the location "start-A" to the location "receive B".

3.5.2 Interaction with other vehicles

As discussed earlier, in our model, the natural platooning of non-vehicles occurs within a small portion of the road that is close to an intersection. The key to modeling such an interaction among multiple (non-bus) vehicles is the use of "Conveyors" (in lieu of "Locations") that are created to represent the lanes, where non-bus vehicles base their movements on FIFO rule in passing the intersection.

In Promodel, a Conveyor is a location that simulates the movement of accumulating or non-accumulating entities. Some of its important properties are summarized below:

- Entities traveling on a conveyor can enter only at the beginning and leave only at the end. Once entities begin movement along a conveyor, they **may not pass each other**. If multiple entities are on a conveyor, only the entity at the end of the conveyor continues to process.

- **When using an accumulating conveyor**, **if the lead entity is unable to exit the conveyor, trailing entities queue behind it.**

These properties explicitly imply the nature of vehicle's platooning, which is, if the leading vehicle in the lane can not exit the intersection due to a red signal or some other reasons, all its followers will have to wait behind it, one after another. Therefore, by specifying the length and speed of the conveyor in the Conveyor/Queue dialog box, the platooning of non-bus vehicles is modeled.

Since a bus has its own lane all the way through the entire thoroughfare, there is no issue of waiting behind non-bus vehicles when traveling along the thoroughfare. However, there does exist the interaction between two buses traveling in the opposite directions. When the intersection has two buses coming from opposite directions (e.g., one from A to C and the other from C to A) at approximately the same time, the one arriving later has to wait until the earlier one has travel past the intersection. The modeling of such an interaction is achieved by Promodel-" IF "logic language, as can be read from the Structure of Model attached in the end of the report.

3.5.3 Interaction with the signal

In absence of a bus, regular signals control vehicle's movements at the intersection by cyclically changing colors between Green, Yellow and Red. A green color means Pass, a red means Stop, and a yellow is a warning sign of the change from a green to a red.

Modeling of the vehicle's interaction with signals can be achieved by using Promodel's logic language after the cyclic color change of regular signal has been defined, as discussed in section 3.4,. The key is the use of Variable - one of the promodel's elements, and the execution of the "IF" logic languages.

*Variables are placeholders for either real or integer numbers that may change during the simulation; Variables are typically **used for making decisions** or for gathering data; Global variables are accessible from anywhere and at any time in the model; The value of a global variable may be displayed dynamically during the simulation and it may even be changed interactively.*

With the above acknowledgement, variables are created to count the number of buses in the system, the number of buses that go in a specific direction of (either going-straight, turning left or turning right), and the number of a specific green in the system. (Recall that, in section 3.4, we introduced four specific greens upon each type of traffic rules. They arrive the intersection in a certain sequence one by one in a cyclic pattern, as can be defined in the arrival table of the model).

The variable named “num-Abus” plays an important role that controls the interface between regular and advanced signals. The approach of a bus can be sensed by the signaling system few seconds (3 seconds in this study) ahead of the bus arrival, triggering the start of he advanced signals. (Modeling the “sensing” is simply done simply assigning a 3 second time in the “move logic” window to make the bus stay in one location before it is routed to the next location). Promodel increases the value of the variable “num-Abus” each time when the system has sensed the presence of a bus, and decreases the value of this variable at the time when the a bus exits the system (i.e., the intersection). A non-zero value of this variable will last until the a bus finishes its travel through the intersection, meaning that the preemption will last as long as this value is non-zero. At the time when the value is changed to zero, the preemption ends.

If the system has no buses in it, i.e., if the variable of num-Abus is zero, at the time when a specific entity of green color arrives in the system, Promodel will increase the number of that green entity to 1, then this green will stay in the system for a known length of time before it is routed to exit the system. At the time when the green finishes its duration i.e., when the green exits the system), Promodel will decrease the value of the variable - the number of that green - to zero. Using the “IF” language of the model in lieu of “locations” where the vehicle (the entity) enters, will make the vehicle move or stop accordingly. In other words, if the number of a specific green is 1 (during the stay of the green in the system), vehicles expecting that green will move to travel through the intersection. Also, if the number of the green is 0 (during the absence of that green in the system), vehicles expecting that green will stop and not travel through the intersection until the system experiences the next arrival of that specific green.

In the presence of a bus, i.e., if the num-Abus is non-zero, the Advanced signals take over the control of movements of all vehicles, preempting the Regular signals regardless which type of the Regular signal is active. The resulting impact of the signal preemption is shown in Table 2. The key to simulate such an impact is also through the use of variables and the “IF” logic language.

Take an example from Table 2. The set of traffic rules governing vehicle movement from A straight through the intersection to C is implemented by the following codes:

```
if num_green_go_AC<1 then

wait until num_green_go_AC=1

else

   begin

   if num_Abus_to_B>0 then

   wait until num_Abus_to_B=0
```

end

move for 2sec

This rule states that, in the absence of a bus, the vehicle can travel through the intersection if the traffic light for AC traffic to go-straight is green; in the presence of a bus, the vehicle will encounter one of the following six possibilities:

1) if a bus coming from A is to go straight (To C) to pass the intersection, then the vehicle **can** also go along with the bus to C

2) if a bus coming from A is to turn to right (To B) to pass the intersection, then the vehicle **can not** move, it must wait in the lane.

3) if a bus coming from A is to turn to left (To D) to pass the intersection, then the vehicle **can** go its way to C

4) if a bus coming from C is to go straight (To A) to pass the intersection, then the vehicle **can** go its way to C

5) if a bus coming from C is to turn to right (To D) to pass the intersection, then the vehicle **can** go its way to C

6) if a bus coming from C is to turn left (To B) to pass the intersection, then the vehicle **can not move,** it must wait in the lane.

In short, a vehicle can only move under two circumstances. One is when the intersection has a regular green (which also means no Abus interferes) that is specifically wanted by the vehicle. The other is when the intersection does not have such a green, but the vehicle can still move while the a bus is moving provided the vehicle's movement does not conflict with that of the bus.

3.6 Modeling of Performance Measures

The performance measure used in this study is the expected amount of time a non-bus vehicle spends on traveling through an intersection. With the idea that simulation itself could generate the amount of time the entity stays in the system, the model is run first, then the model output could directly tell the time the entity is in system , which is equal to tell the amount of time a non-bus vehicle spends in passing the intersection.

Table 3: Parameters and Assumptions As Initial Model Input

Ref #	Description of Parameters	Assumptions	Unit of Measurement	Value as Initial Model Input
1	interarrival time the conventional vehicles arrive at the intersection (Table 4)	The arrival rate is a random variable distributed exponentially	seconds	E(2) x E(2) main x main intersection E(2) x E(8) for main x secondary
2	cycle time of signal change (Table 5)	Deterministic,	seconds	72 for main x main intersection 60 for main x secondary intersection
3	Duration of a green signal (Table 5)	Deterministic	seconds	16 x 16 for main x main intersection 16 x 10 for main x secondary
4	Duration of a red signal (Table 5)	Deterministic	seconds	54 x 54 for main x main intersection 42 x 48 for main x secondary
5	Duration of a yellow signal between each change from green to red (Table5)	Deterministic	seconds	2 seconds for all cases

6	The amount of time the vehicle spends in the movement to pass the intersection (either turn left, or go-through, or turn right)	Deterministic	seconds	2
7	The probability that the vehicle would stay with the current traffic	Deterministic		table 6
8	The probability that the vehicle would turn to left (entering the cross traffic)	Deterministic		table 6
9	The probability that the vehicle would turn to right (entering the cross traffic)	Deterministic		table 6
10	The interarrival time of the ABUS at the intersection	Normal distribution	minutes	Step 1: N(5,1.5) Step 2: N(10,1) Step 3: N(15,0.5)
11	The amount of time the ABUS spends in the movement of passing the intersection (either turn left, or go-through, or turn right)	Deterministic	seconds	5
12	All vehicle's travel time between two adjacent intersection	A Normal random variable, based on the assumption that all vehicles are assumed to travel at a constant speed	minutes	From Intersection 1 to 2 is N(5,0.5) From Intersection 2 to 3 is N(4.5,0.45) From Intersection 3 to 4 is N(6,0.6)

Remarks :

1. The relationship of parameters 2,3,4,5 is as follows.

For type 1 conventional traffic lights,

green for AC/CA flows to go straight + yellow + green for AC/CA flows to turn to left + yellow + green for BD/DB flows to go straight + yellow + green for BD/DB flows to turn to left + yellow = the cycle time

or

a green + a yellow + a red = the cycle time

For type 2 conventional traffic lights:

Green for the flow originated from A + yellow + Green for the flow originated from B + yellow + Green for the flow originated from C + yellow + Green for the flow originated from D + yellow = the cycle time

Or

a green + a yellow + a red = the cycle time

2. Four consecutive intersections are connected as a hypothetical section of corridor for simulation study.

Table 4. Vehicle's Arrival and Movement Time as initial model input

	ABUS interarrival time (in minute)	ABUS moving time (in seconds)	conventional vehicle interarrival time (in seconds)				vehicle moving time (in seconds)
			main x main intersection		Main x secondary intersection		
			Through traffic (main)	cross traffic (main)	Through (main)	cross traffic (secondary)	
Step 1	N(5,1.5)	5	E(2)	E(2)	E(2)	E(8)	2
Step 2	N(10,1)	5	E(2)	E(2)	E(2)	E(8)	2
Step 3	N(15,0.5)	5	E(2)	E(2)	E(2)	E(8)	2

Table 5. Quantitative Distribution of Color Signals as initial model input (time in seconds)

	At main x main Intersection (Type A)		At main x secondary Intersection(Type B)	
	corridor (main)	cross traffic (main)	corridor (main)	cross traffic (secondary)
frequency per cycle	72	72	60	60
duration of a green	16	16	16	10
Duration of a red	54	54	42	48
yellow between each green and red	2	2	2	2
Note that for both Type A and B intersection, the duration of a yellow color of signals is set up same (two seconds)				

Table 6. Probabilistic Distribution of Vehicle Movement at the intersection

	Current state (position) of vehicles			
	Intersection type			
	main x main		main x secondary	
Possible Vehicle's Movement	on the corridor (main)	on the cross traffic (main)	on the corridor (main)	on the cross traffic (secondary)
go straight	0.5	0.5	0.8	0.2
turn left	0.25	0.25	0.1	0.4
turn right	0.25	0.25	0.1	0.4
Remarks: The probabilities applied to the ABUS as well.				

4. Experiments of Simulation Scenarios

Five models have been developed using the Promodel simulation model. All models are centered around our objective to measure the amount of time a non-bus vehicle spends for passing the intersection under different intermediate steps toward the ABUS concepts. Formatted listings of three of the models are provided in the Appendix.

Each and every model represents a certain combination of physical and hypothetical traffic conditions possessed by an intersection, as discussed in previous sections. We experiment each with four simulation scenarios. The four scenarios correspond to the base case where there is no bus lane and no transit signal priority/preemption, the first two intermediate steps toward an ABUS (i.e., Step 1 and Step 2), and an ABUS (i.e., either Step 3 and Step 4). The three operational steps is compared to the base in order to investigate how differently each step would affect those non-bus vehicles in their movement of passing one or more intersections.

The five models are put in two groups. Group 1 simulation contains four models (Model 1, 2, 3, 4), in which a single intersection is the subject under study. Each and every operational step is simulated four times across these four models for a total of sixteen combinations of traffic conditions of an intersection, as shown in Table 7.

Group 2 simulation contains one model (model 5). The model is an extension of individual models of group 1, which links four consecutive intersections to represent a hypothetical section of a thoroughfare. Although the model was purely developed to study the traffic flow through multiple intersections, the model also supplementally verified the results of group 1 simulations. We selected one combination of traffic conditions and experimented with the four simulation scenarios as well, as shown in Table 8.

Note that technically, for each and every model, different scenarios are simulated through changing the value of the parameters that can be defined in the arrival table of the Promodel. For instance, to simulate the base case, we only need to disable the function of the bus-related elements in the arrival table of the simulation model.

5. SUMMARY AND ANALYSIS OF RESULTS

Parameters assumed and their corresponding values as model specifications are given in Table 3,4,5,6.

For all models, the system is modeled as a non-terminating system. After truncating 60 minutes of data representing transient start-up effect, an 8-hour worth of data is collected. The simulation of the base case is first to be run for each model. Results of all runs are summarized in Table 9 and Table10. The statistic collected of each run is the average time (in minutes) a non-bus vehicle spends in passing the intersection, tabulated in Table 9 and Table 10, for group 1 and group 2 simulations respectively.

As can be seen from Table 9, at a 95% confidence, the mean time a conventional vehicle spends to pass the intersection increases when transit signal priority/preemption is implemented, compared with the base case. However, this increase decreases as the operation is improved to a higher operational step.

On average, for all selected combinations of intersections and traffic signaling configurations, it would result in an overall **14.40 %** increase of disturbance to the cross traffic and **15.66 %** to the through traffic if Step 1 toward ABUS is implemented, a **7.05%** increase of disturbance to the cross traffic and **7.91 %** to the through traffic if Step 2 toward ABUS is implmented, and **4.21%** increase of disturbance to the cross traffic and **5.19 %** to the through traffic if ABUS is implemented (i.e., either Step 3 or Step 4).

It's interesting to see that, unlike it was thought to be, the disturbance caused by the ABUS signal preemption to the cross traffic is not greater than it is to the through traffic. The traffics on both streets appear to be affected not much differently on average.

In general, the cross traffic at a main x main intersection tends to be more disturbed than that of a main x secondary intersection, either with or without implementing ABUS or its precursor Steps.

In comparing the results from two simulation groups, it turns out that both groups of models agree with each other in the sense of the disturbance on the cross traffic. Both groups of simulation yield the same information in that, the disturbance on the cross traffic, under Step 1 is about 14% higher than that of the base case, under Step 2 about 7% and under Step 3 about 4% .

Table 7. Combinations of Traffic Scenarios used to run experiments for Group 1 simulation

Scenario Ref Number		ABUS operation	Traffic Light	Intersection Type
Model 1	Base	Existing system	type1	main x main (TypeA)
	#1.	step 1	type1	main x main (Type A)
	#2.	step 2	type1	main x main (Type A)
	#3.	step 3	type1	main x main (Type A)
Model 2	Base	Existing system	type1	main x secondary (Type B)
	#4.	step 1	type1	main x secondary (Type B)
	#5	step 2	type1	main x secondary (Type B)
	#6.	step 3	type1	main x secondary (Type B)
Model 3	Base	Existing system	type2	main x main (TypeA)
	#7	step 1	type2	main x main (TypeA)
	#8	step 2	type2	main x main (TypeA)
	#9	step 3	type2	main x main (TypeA)
Model 4	Base	Existing system	type2	main x secondary (Type B)
	#10	step 1	type2	main x secondary (Type B)
	#11	step 2	type2	main x secondary (Type B)
	#12	step 3	type2	main x secondary (Type B)
Remarks: scenario #1, 4,7,10 are to simulate ABUS operation step 1 scenario #2, 5,8,11 are to simulate ABUS operation step 2 scenario #3,6,9,12 are to simulate ABUS operation step 3				

Table 8. Combinations of Traffic Scenarios used to Run Experiments for Group 2 Simulation

	ABUS operation	Type of regular Traffic Signals	Intersection 1 (origin/destination)	Intersection 2	Intersection 3	Intersection 4 (origin/destination)
Model 5	Base	Existing system	Main x main	Main x secondary	Main x secondary	Main x main
	Step 1	Type 1 traffic signals/rules	Main x main	Main x secondary	Main x secondary	Main x main
	Step 2	Type 1 traffic signals/rules	Main x main	Main x secondary	Main x secondary	Main x main
	Step 3	Type 1 traffic signals/rules	Main x main	Main x secondary	Main x secondary	Main x main

Remarks:

a. Only type 1 traffic signals/rules is studied in this model

b. For AC traffic flow, the intersection 1 is considered the origin where the ABUS accesses the corridor , and the intersection 4 is the destination where the Abus egresses the corridor; Inversely, for CA traffic flow, the intersection 4 is considered the origin where the ABUS accesses the corridor and the intersection 1 is the destination where the Abus egresses the corridor. It is assumed in this model that the Abus does not make any entry or exit inbetween.

.

6. CONCLUSION AND RECOMMENDATIONS

Five simulation models in two groups have been built and the associated parameters are assumed on basis of the real world abstractions and observations. The objective of this study has been to use computer simulation to justify if the deployment of the ABUS operation concept would decrease the disturbance caused by signal preemptions to the surrounding traffic, in particular to the cross traffic; and how much difference would each intermediate step toward an ABUS system make in reducing the disturbance. The study shows that the disturbance does increase considerably above the base case when the transit signal priority is deployed. The study also shows that improving the operation with ITS to cluster buses into convoys or with automation to organize buses to closely-spaced convoys would reduce the growth of the disturbance to a certain degree.

The result of this research provides a quantitative assessment of the positive effects of ITS and automation technologies on reducing the disturbance of transit signal priority on surrounding traffic of an intersection of two city streets. It is recommended that the methodology of factorial design experiments be used for further study to verify the results. Representation of the four Steps as scenarios to be modeled and simulated with ProModel can be improve to provide more accurate assessments. Possible considerations for further study might include:

- Consider the variability of buses' moving time through the intersection. In the current simulation model, a fixed moving time is used.
- Impose traffic rules to constrain right-turn movement when no buses are present at the intersection
- Simulate natural platooning and passing of vehicles between two intersections.
- Simulate travel time for bus routes connecting off-mainline origins and destinations.
- Simulate the tradeoff between the reduction of disturbance through manual clustering or automated closely-spaced convoying and the resulting longer passenger travel time due to the need for buses to wait for other buses to form clusters or convoys.

REFERENCES

Carey, G. N., Viggiano, S. M., and Gardner, L., "The development of a guided bus system: the Eugene/Springfield experience," Institute of Transportation Engineers. Meeting (68th : 1998 : Toronto, Ont.). Annual meeting papers, Institute of Transportation Engineers, Washington DC.

Chen, C.-Y. and Litkouhi, B., "Technological Challenges in the Development of AHS," ITS Journal, Vol. 4, No. 1-2, pp. 81-100, 1998.

DeMarco, V. R., "Dual-Mode System Development Program of the Urban Mass Transportation Administration," Proceedings of International Conference on Dual Mode Transportation, Washington, D.C., 1974; Transportation Research Board Special Report No. 170, **:** Transportation Research Board, National Research Council, Washington, D.C..

Hall, R.W., Nowroozi, A., and Tsao, H.-S.J., "Entrance Capacity of an Automated Highway System," to *Transportation Science*, Vol. 35, No. 1, pp. 19 – 36, 2001.

Hall, R.W., and H.-S. J. Tsao "Capacity of Automated Highway Systems: Merging Efficiency," Proceedings of American Control Conference, Albuquerque, New Mexico, June, 1997, pp. 2046-2050.

Hanson, M. and H.-S.J. Tsao, "Leveraging Exogenous Events for the Deployment of Automated Highway Systems," Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida (1996).

Jacobson, J. and Sheffi, Y., "Analytical Model for Traffic Delays Under Bus Signal Preemption: Theory and Application," Transportation Research B, Vol. 15B, 1981.

Sunkari, S.R., Beasley, P.S. Urbanik II, T., Fambro, D.B., "Model to Evaluate the Impacts of Bus Priority on Signalized Intersections," Transportation Research Board 1494, 1995.

Tsao, H.-S. J. (1995a), "Traffic Control for Automated Highway Systems: A Conceptual Framework", Transportation Research, Part C, Vol. 3, No. 4, pp. 227-246, 1995.

Tsao, H.-S. J. (1995b), "Stage Definition for AHS Deployment and an AHS Evolutionary Scenario", Intelligent Vehicle Highway Systems Journal, Vol.2(4), pp. 359-382, 1995.

Tsao, H.-S. J. (1995c), "A Staggered-Diamond Design for Automated/Manual-HOV Highway-to-Highway Interchanges and Constraints on AHS Design for Accommodating Automated Highway Change", Intelligent Vehicle Highway Systems Journal, Vol.2(3), pp. 281-292, 1995.

Tsao, H.-S. J. (1995d), "Constraints on Initial AHS Deployment and the Concept Definition of a Shuttle Service for AHS Debut", IVHS Journal, Vol. 2(2), 159-173, 1995.

Tsao, H.-S. J. (1998a), "From the Guest Editor," Special Issue of the ITS Journal on Automated Highway Systems, Vol. 4, Nos:1-2, 1998.

Tsao, H.-S.J. (1998b), "An Axiomatic Approach to Developing Partial Automation Concepts for Deployment of Automated Highway Systems and Partial Invocation of Vision-Based Lane-Keeping and Adaptive Cruise Control," *Transportation Research Record*, No. 1651, 1998.

Tsao, H.-S. J. (1998c), "The Role of Air Cargo in California's Goods Movement,", UCB-ITS Research Report UCB-ITS-RR-98-7, University of California at Berkeley, Institute of Transportation Studies, Sept. 1998.

Tsao, H.-S. J. (2001), " A Framework for Evaluating Deployment Strategies for Intelligent Transportation Systems", Intelligent Transportation Systems Journal (ITS Journal), Vol. 6, pp. 141-173, 2001.

Tsao, H.-S.J. and Botha, J.L. (2001), "Operating Concepts for Urban Bus Automation and Inter-city Truck Automation," the Proceedings of the 2001 IEEE ITS Conference, Oakland, California.

Tsao, H.-S.J. and Botha, J.L (2002a) , "An Automated Highway System Dedicated To Inter-City Trucking: Operating Concepts And Deployment," presented at the 2002 Annual Meeting of the Transportation Research Board on Jan. 14, 2002, Washington, D.C.

Tsao, H.-S.J. and Botha, J.L. (2002b), "An Automated Highway System Dedicated To Inter-City Trucking: Design Options, Operating Concepts And Deployment,", Intelligent Transportation Systems Journal (The ITS Journal), Vol. 7, No. 2, pp. 169 – 196, 2002.

Tsao, H.-S. J., Botha, J.L. (2003), "Definition and Evaluation of Bus and Truck Automation Operations Concepts: Final Report," California PATH Research Report UCB-ITS-PRR-2003-19, May 2003.

Tsao, H.-S.J. and Ran, B, "Driver Intelligence Replacement in a Decision-Oriented Deployment Framework for Driving Automation",., *Proceedings of The Third World Congress on Intelligent Transportation Systems*, Orlando, Florida (1996).

Tsao, H.-S. J. and Hall, R.W., "A Probabilistic Model for AVCS Longitudinal Collision/Safety Analysis", Intelligent Vehicle Highway Systems Journal, Vol.1, No. 3, pp. 261-274, 1994.

Tsao, H.-S. J., Hall, R.W., and Chatterjee, I., "Analytical Models for Vehicle/Gap Distribution on Automated Highway Systems", Transportation Science, Vol. 31, No. 1., pp. 18-33, 1997.

Tsao, H.-S. J., R.W. Hall and S.E.Shladover, "Design Options for Operating Automated Highway Systems", Proceedings of Vehicle Navigation & Information Systems Conference (Oct. 1993).

Tsao, H.-S.J., Hall, R.W., Shladover, S.E., Plocher, T.A. and Levitan, L.J., "Human Factors Design of Automated Highway Systems: First Generation Scenarios," US-DOT FHWA Report FHWA-RD-93-123 (1993).

Tsao, H.-S. J.,, T.A. Plocher, W.B. Zhang, and S.E. Shladover, "Human Factors Design for Automated Highway Systems: Second Generation Scenarios", draft US-DOT FHWA Report, under FHWA Contract No. DTFH61-91-C-00100, (Jan. 14, 1994).

Tsao, H.-S. J. and Ran, B., "Driver Intelligence Replacement in a Decision-Oriented Deployment Framework for Driving Automation", Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida (1996).

Table 9 : The effect of ABUS operation concepts to the surrounding traffic-result of Group 1 simulation model (time in minutes)

				Through Traffic				**Cross Traffic**			
	ABUS operation	Type of Traffic lights	Intersection type	Vehicle from A	Vehicle from C	Average % disturbance increase	Average % disturbance reduction of a higher step	Vehicle from B	Vehicle from D	Average % disturbance increase	Average % disturbance reduction of higher step
	without ABUS			**0.3303**	**0.328**	**baseline**		**0.3272**	**0.3273**	**baseline**	
#1.	step 1	type1	main x main	0.374	0.3754	13.8408%	baseline	0.3805	0.3828	16.6233%	baseline
#2.	step 2	type1	main x main	0.3533	0.3521	7.1555%	5.8707%	0.3537	0.354	8.1283%	7.2
#3.	step 3	type1	main x main	0.3474	0.3461	5.3477%	7.4587%	0.3421	0.3409	4.3545%	10.5
	without ABUS			**0.303**	**0.3018**	**baseline**		**0.2574**	**0.2555**	**baseline**	
#4.	step 1	type1	main x secondary	0.3516	0.3536	16.6016%	baseline	0.2874	0.288	12.1876%	baseline
#5	step 2	type1	main x secondary	0.3272	0.3289	8.4831%	6.9625%	0.2699	0.2684	4.9526%	6.4
#6.	step 3	type1	main x secondary	0.3202	0.3199	5.8370%	9.2306%	0.2629	0.2639	2.7122%	8.4
	without ABUS			**0.3289**	**0.3265**	**baseline**		**0.3288**	**0.3273**	**baseline**	
#7	step 1	type2	main x main	0.3741	0.3854	15.8913%	baseline	0.3747	0.3795	14.9543%	baseline
#8	step 2	type2	main x main	0.3502	0.3552	7.6332%	7.1123%	0.3605	0.3494	8.1967%	5.8
#9	step 3	type2	main x main	0.345	0.341	4.6681%	9.6496%	0.3481	0.3389	4.7070%	8.8
	without ABUS			**0.3023**	**0.3027**	**baseline**		**0.2606**	**0.2583**	**baseline**	
#10	step 1	type2	main x secondary	0.3531	0.3506	16.3144%	baseline	0.2978	0.2929	13.8350%	baseline
#11	step 2	type2	main x secondary	0.3261	0.3295	8.3633%	6.8324%	0.2778	0.2771	6.9393%	6.0
#12	step 3	type2	main x secondary	0.3192	0.3156	4.9261%	9.7918%	0.271	0.2741	5.0539%	7.7
				step 1		15.6620%		step 1		14.4000%	
Overall effect of ABUS operation concept				step 2		7.9088%		step 2		7.0542%	
				step 3		5.1947%		step 3		4.2069%	

Table 10: The effect of ABUS operation to the cross traffic -result of Group 2 simulation (time in minutes)

	intersection 1 (origin/destination)		intersection 2		intersection 3		intersection 4 (destination/ origin)		overall % disturbance increase
ABUS operation	mean time to pass through	avg % disturbance increase	mean time to pass through	avg % disturbance increase	mean time to pass through	avg % disturbance increase	mean time to pass through	avg % disturbance increase	
without ABUS	0.424	baseline	0.3905	baseline	0.3966	baseline	0.4227	baseline	base
step 1	0.4814	13.54%	0.4526	15.90%	0.4486	13.11%	0.4816	13.93%	14.12%
step 2	0.4537	7.00%	0.427	9.35%	0.4169	5.12%	0.454	7.40%	7.22%
step 3	0.4407	3.94%	0.4118	5.45%	0.4121	3.91%	0.445	5.28%	4.64%

Appendix A: The Structure of Model 1 – Simulation of a Main x Main Intersection with Type 1 Traffic Light Control

```
********************************************************************************
*                                                                              *
*                          Formatted Listing of Model 1:                       *
*            C:\ProMod4\models\masterproj\proj1-type1-mainxmain-g.MOD          *
*                                                                              *
********************************************************************************

  Time Units:                        Minutes
  Distance Units:                    Feet


********************************************************************************
*                                  Locations                                   *
********************************************************************************

  Name         Cap Units Stats       Rules              Cost
  ------------ --- ----- ----------- ------------------ ------------
  A1_car_input inf 1     Time Series Oldest, ,
  A1_L_queue   inf 1     Time Series Oldest, ,
  A1_G_queue   inf 1     Time Series Oldest, ,
  A1_R_queue   inf 1     Time Series Oldest, ,
  A1_L_lane    5   1     Time Series Oldest, FIFO,
  A1_G_lane    5   1     Time Series Oldest, FIFO,
  A1_R_lane    5   1     Time Series Oldest, FIFO,
  B1_car_input inf 1     Time Series Oldest, ,
  B1_L_queue   inf 1     Time Series Oldest, ,
  B1_G_queue   inf 1     Time Series Oldest, ,
  B1_R_queue   inf 1     Time Series Oldest, ,
  B1_L_lane    5   1     Time Series Oldest, FIFO, First
  B1_G_lane    5   1     Time Series Oldest, FIFO,
  B1_R_lane    5   1     Time Series Oldest, FIFO,
  C1_car_input inf 1     Time Series Oldest, ,
  C1_L_queue   inf 1     Time Series Oldest, ,
  C1_G_queue   inf 1     Time Series Oldest, ,
  C1_R_queue   inf 1     Time Series Oldest, ,
  C1_L_lane    5   1     Time Series Oldest, FIFO,
  C1_G_lane    5   1     Time Series Oldest, FIFO,
  C1_R_lane    5   1     Time Series Oldest, FIFO,
  D1_car_input inf 1     Time Series Oldest, ,
  D1_L_queue   inf 1     Time Series Oldest, ,
  D1_G_queue   inf 1     Time Series Oldest, ,
  D1_R_queue   inf 1     Time Series Oldest, ,
  D1_L_lane    5   1     Time Series Oldest, FIFO,
```

```
  D1_G_lane    5   1     Time Series Oldest, FIFO,
  D1_R_lane    5   1     Time Series Oldest, FIFO,
  Abus_lane_AC inf 1     Time Series Oldest, FIFO,
  intersection inf 1     Time Series Oldest, ,
  left         inf 1     Time Series Oldest, ,
  go           inf 1     Time Series Oldest, ,
  right        inf 1     Time Series Oldest, ,
  to_A         inf 1     Time Series Oldest, ,
  to_B         inf 1     Time Series Oldest, ,
  to_C         inf 1     Time Series Oldest, ,
  to_D         inf 1     Time Series Oldest, ,
  abus_lane_CA inf 1     Time Series Oldest, FIFO,
  sensor_A     1   1     Time Series Oldest, ,
  sensor_C     1   1     Time Series Oldest, ,


********************************************************************************
*                                   Entities                                   *
********************************************************************************

  Name             Speed (fpm)  Stats       Cost
  ---------------- ------------ ----------- ------------
  Car_A            150          Time Series
  Car_B            150          Time Series
  Car_C            150          Time Series
  Car_D            150          Time Series
  ABUS_A           0            Time Series
  ABUS_C           0            Time Series
  green_left_AC    150          Time Series
  green_left_BD    150          Time Series
  green_go_AC      150          Time Series
  green_go_BD      150          Time Series
  dummy_green_go   150          Time Series
  dummy_green_left 150          Time Series


********************************************************************************
*                                  Processing                                  *
********************************************************************************

                                     Process                               Routing

 Entity        Location     Operation            Blk  Output           Destination  Rule       Move Logic
 ------------- ------------ ------------------   ---- ---------------- ------------ ----------  ------------
 green_go_AC   intersection if num_abus>0 then
                            route 1
                            else
```

```
                             begin
                             inc num_green_go1
                             route 2
                             end




                                               1    dummy_green_go   EXIT         FIRST 1
                                               2    green_go_AC      go           FIRST 1
green_go_AC   go           wait 16sec          1    green_go_AC      EXIT         FIRST 1     dec num_green_go1
green_left_AC intersection if num_abus>0 then
                           route 1
                           else
                              begin
                              inc num_green_left1
                              route 2
                              end


                                               1    dummy_green_left EXIT         FIRST 1
                                               2    green_left_AC    left         FIRST 1
green_left_AC left         wait 16sec          1    green_left_AC    EXIT         FIRST 1     dec num_green_left1
green_go_BD   intersection if num_abus>0 then
                           route 1
                           else
                              begin
                              inc num_green_go2
                              route 2
                              end              1    dummy_green_go   EXIT         FIRST 1
                                               2    green_go_BD      go           FIRST 1
green_go_BD   go           wait 16sec          1    green_go_BD      EXIT         FIRST 1     dec num_green_go2
green_left_BD intersection if num_abus>0 then
                           route 1
                           else
                              begin
                              inc num_green_left2
                              route 2
                              end              1    dummy_green_left EXIT         FIRST 1
                                               2    green_left_BD    left         FIRST 1
green_left_BD left         wait 16sec          1    green_left_BD    EXIT         FIRST 1     dec num_green_left2
ABUS_A        intersection time1 = clock()     1    ABUS_A           Abus_lane_AC FIRST 1
```

```
ABUS_A        Abus_lane_AC




                                    1    ABUS_A          sensor_A     FIRST 1

ABUS_A        sensor_A     inc num_abus

                           if num_abus_C>0 then
                           wait until num_abus_C=0

                           inc num_abus_A
                                    1    ABUS_A          to_D,500     0.250000 1  time1 = clock() - time1
                                                                                  var1 = time1
                                                                                  move for 3sec

                                                         to_C,500     0.500000    time1 = clock() - time1
                                                                                  var2 = time1
                                                                                  move for 3sec


                                                         to_B,500     0.250000    time1 = clock() - time1
                                                                                  var3 = time1
                                                                                  move for 3sec

ABUS_A        to_D         inc num_abus_to_D
                           wait 5sec

                           dec num_abus_to_D
                           dec num_abus_A
                           dec num_abus         1    ABUS_A          EXIT         FIRST 1

ABUS_A        to_C         inc num_abus_to_C
                           wait 5sec

                           dec num_abus_to_C
                           dec num_abus_A
                           dec num_abus         1    ABUS_A          EXIT         FIRST 1
```

ABUS_A	to_B	inc num_abus_to_B wait 5sec dec num_abus_to_B dec num_abus_A dec num_abus	1	ABUS_A	EXIT	FIRST 1	
Car_A	intersection		1	Car_A	A1_car_input	FIRST 1	
Car_A	A1_car_input		1	Car_A	A1_L_queue,1 A1_G_queue,1 A1_R_queue,1	0.250000 1 0.500000 0.250000	
Car_A	A1_L_queue		1	Car_A	A1_L_lane	FIRST 1	
Car_A	A1_G_queue		1	Car_A	A1_G_lane	FIRST 1	
Car_A	A1_R_queue		1	Car_A	A1_R_lane	FIRST 1	
Car_A	A1_L_lane	time_a=clock()	1	Car_A	to_D	FIRST 1	if num_green_left1<1 then wait until num_green_left1=1 move for 2sec time_a=clock()-time_a var=time_a

Car_A	A1_G_lane		1	Car_A	to_C	FIRST 1	if num_green_go1<1 then
							wait until num_green_go1=1
							else
							  begin
							  if num_abus_to_B>0 then
							  wait until num_abus_to_B=0
							  end
							move for 2sec

Car_A	A1_R_lane		1	Car_A	to_B	FIRST 1	if num_abus_to_B>0 then wait until num_abus_to_B=0 move for 2sec

```
Car_B         intersection                      1    Car_B            B1_car_input FIRST 1
Car_B         B1_car_input                      1    Car_B            B1_L_queue   0.250000 1
                                                                      B1_G_queue   0.500000
                                                                      B1_R_queue   0.250000
Car_B         B1_L_queue                        1    Car_B            B1_L_lane    FIRST 1
Car_B         B1_G_queue                        1    Car_B            B1_G_lane    FIRST 1
Car_B         B1_R_queue                        1    Car_B            B1_R_lane    FIRST 1
Car_B         B1_L_lane                         1    Car_B            to_A         FIRST 1     if num_green_left2<1 then
                                                                                               wait until num_green_left2>0
                                                                                               move for 2sec

Car_B         B1_G_lane                         1    Car_B            to_D         FIRST 1     if num_green_go2<1 then
                                                                                               wait until num_green_go2>0
                                                                                               move for 2sec


Car_B         B1_R_lane                         1    Car_B            to_C         FIRST 1     move for 2sec
Car_C         intersection                      1    Car_C            C1_car_input FIRST 1
Car_C         C1_car_input                      1    Car_C            C1_L_queue   0.250000 1
                                                                      C1_G_queue   0.500000
                                                                      C1_R_queue   0.250000
Car_C         C1_L_queue                        1    Car_C            C1_L_lane    FIRST 1
Car_C         C1_G_queue                        1    Car_C            C1_G_lane    FIRST 1
Car_C         C1_R_queue                        1    Car_C            C1_R_lane    FIRST 1
Car_C         C1_L_lane                         1    Car_C            to_B         FIRST 1     if num_green_left1<1 then
                                                                                               wait until num_green_left1>0
                                                                                               move for 2sec


Car_C         C1_G_lane                         1    Car_C            to_A         FIRST 1     if num_green_go1<1 then
                                                                                               wait until num_green_go1=1
                                                                                               else
                                                                                                 begin
                                                                                                 if num_abus_to_D>0 then
                                                                                                 wait until num_abus_to_D=0
                                                                                                 end
                                                                                               move for 2sec

Car_C         C1_R_lane                         1    Car_C            to_D         FIRST 1     if num_abus_to_D>0 then
                                                                                               wait until num_abus_to_D=0
                                                                                               move for 2sec


Car_D         intersection                      1    Car_D            D1_car_input FIRST 1
Car_D         D1_car_input                      1    Car_D            D1_L_queue   0.250000 1
                                                                      D1_G_queue   0.500000
                                                                      D1_R_queue   0.250000
```

Car_D	D1_L_queue		1	Car_D	D1_L_lane	FIRST 1		
Car_D	D1_G_queue		1	Car_D	D1_G_lane	FIRST 1		
Car_D	D1_R_queue		1	Car_D	D1_R_lane	FIRST 1		
Car_D	D1_L_lane		1	Car_D	to_C	FIRST 1		if num_green_left2<1 then wait until num_green_left2>0 move for 2sec
Car_D	D1_G_lane		1	Car_D	to_B	FIRST 1		if num_green_go2<1 then wait until num_green_go2>0 move for 2sec
Car_D	D1_R_lane		1	Car_D	to_A	FIRST 1		move for 2sec
ALL	to_A		1	ALL	EXIT	FIRST 1		
ALL	to_B		1	ALL	EXIT	FIRST 1		
ALL	to_C		1	ALL	EXIT	FIRST 1		
ALL	to_D		1	ALL	EXIT	FIRST 1		
ABUS_C	intersection		1	ABUS_C	abus_lane_CA	FIRST 1		
ABUS_C	abus_lane_CA		1	ABUS_C	sensor_C	FIRST 1		
ABUS_C	sensor_C	inc num_abus if num_abus_A>0 then wait until num_abus_A=0 inc num_abus_C	1	ABUS_C	to_B	0.250000	1	move for 3sec
					to_A	0.500000		move for 3sec
					to_D	0.250000		move for 3sec
ABUS_C	to_B	inc num_abus_to_B wait 5sec dec num_abus_to_B dec num_abus_C dec num_abus						

```
                                           1    ABUS_C          EXIT         FIRST 1
 ABUS_C        to_A         inc num_abus_to_A
                            wait 5sec

                            dec num_abus_to_A
                            dec num_abus_C
                            dec num_abus         1    ABUS_C          EXIT         FIRST 1
 ABUS_C        to_D         inc num_abus_to_D
                            wait 5sec

                            dec num_abus_to_D
                            dec num_abus_C
                            dec num_abus         1    ABUS_C          EXIT         FIRST 1

*******************************************************************************
*                                  Arrivals                                   *
*******************************************************************************

  Entity        Location     Qty each   First Time Occurrences Frequency   Logic
  ------------- ------------ ---------- ---------- ----------- ----------- ------------
  Car_A         intersection 1          0          inf         E(2)sec
  Car_B         intersection 1          0          INF         E(2)sec
  Car_C         intersection 1          0          INF         E(2)sec
  Car_D         intersection 1          0          INF         E(2)sec
  ABUS_A        intersection 1          0          INF         N(5,1.5)min

  ABUS_C        intersection 1          2.5min     INF         N(5,1.5)min
  green_go_AC   intersection 1          0          INF         72sec




  green_left_AC intersection 1          18sec      INF         72sec


  green_go_BD   intersection 1          36sec      INF         72sec
  green_left_BD intersection 1          54sec      INF         72sec


*******************************************************************************
*                                 Attributes                                  *
*******************************************************************************

  ID             Type         Classification
  -------------- ------------ --------------
  time1          Real         Entity
```

```
  time2          Real         Entity
  Time3          Real         Entity
  num_abus_left  Integer      Entity
  num_abus_go    Integer      Entity
  num_abus_right Integer      Entity
  time_a         Real         Entity


********************************************************************************
*                              Variables (global)                              *
********************************************************************************

  ID              Type         Initial value Stats
  --------------- ------------ ------------- -----------
  num_abus        Integer      0             Time Series
  num_abus_A      Integer      0             Time Series
  num_abus_C      Integer      0             Time Series
  num_abus_to_D   Integer      0             Time Series
  num_abus_to_C   Integer      0             Time Series
  num_abus_to_B   Integer      0             Time Series
  num_abus_to_A   Integer      0             Time Series
  num_green_go1   Integer      0             Time Series
  num_green_go2   Integer      0             Time Series
  num_green_left1 Integer      0             Time Series
  num_green_left2 Integer      0             Time Series
  var1            Real         0             Time Series
  var2            Real         0             Time Series
  var3            Real         0             Time Series
  var             Real         0             Time Series
```

Appendix B: The Structure of Model 4 – Simulation of a Main x Secondary Intersection with Type 2 Traffic Light Control

```
********************************************************************************
*                                                                              *
*                          Formatted Listing of Model 4:                       *
*         C:\ProMod4\models\masterproj\proj1-type2-mainxsecondary-g.MOD        *
*                                                                              *
********************************************************************************

  Time Units:                       Minutes
  Distance Units:                   Feet


********************************************************************************
*                                  Locations                                   *
********************************************************************************

  Name         Cap Units Stats       Rules              Cost
  ------------ --- ----- ----------- ------------------ ------------
  A1_car_input inf 1     Time Series Oldest, ,
  A1_L_queue   inf 1     Time Series Oldest, ,
  A1_G_queue   inf 1     Time Series Oldest, ,
  A1_R_queue   inf 1     Time Series Oldest, ,
  A1_L_lane    5   1     Time Series Oldest, FIFO,
  A1_G_lane    5   1     Time Series Oldest, FIFO,
  A1_R_lane    5   1     Time Series Oldest, FIFO,
  B1_car_input inf 1     Time Series Oldest, ,
  B1_L_queue   inf 1     Time Series Oldest, ,
  B1_G_queue   inf 1     Time Series Oldest, ,
  B1_R_queue   inf 1     Time Series Oldest, ,
  B1_L_lane    5   1     Time Series Oldest, FIFO, First
  B1_G_lane    5   1     Time Series Oldest, FIFO,
  B1_R_lane    5   1     Time Series Oldest, FIFO,
  C1_car_input inf 1     Time Series Oldest, ,
  C1_L_queue   inf 1     Time Series Oldest, ,
  C1_G_queue   inf 1     Time Series Oldest, ,
  C1_R_queue   inf 1     Time Series Oldest, ,
  C1_L_lane    5   1     Time Series Oldest, FIFO,
  C1_G_lane    5   1     Time Series Oldest, FIFO,
  C1_R_lane    5   1     Time Series Oldest, FIFO,
  D1_car_input inf 1     Time Series Oldest, ,
  D1_L_queue   inf 1     Time Series Oldest, ,
  D1_G_queue   inf 1     Time Series Oldest, ,
  D1_R_queue   inf 1     Time Series Oldest, ,
  D1_L_lane    5   1     Time Series Oldest, FIFO,
  D1_G_lane    5   1     Time Series Oldest, FIFO,
```

```
  D1_R_lane    5   1     Time Series Oldest, FIFO,
  Abus_lane_AC inf 1     Time Series Oldest, FIFO,
  intersection inf 1     Time Series Oldest, ,
  pass         inf 1     Time Series Oldest, ,
  to_A         inf 1     Time Series Oldest, ,
  to_B         inf 1     Time Series Oldest, ,
  to_C         inf 1     Time Series Oldest, ,
  to_D         inf 1     Time Series Oldest, ,
  abus_lane_CA inf 1     Time Series Oldest, FIFO,
  sensor_A     1   1     Time Series Oldest, ,
  sensor_C     1   1     Time Series Oldest, ,


********************************************************************************
*                                   Entities                                   *
********************************************************************************

  Name              Speed (fpm)  Stats       Cost
  ---------------- ------------ ----------- ------------
  Car_A             150          Time Series
  Car_B             150          Time Series
  Car_C             150          Time Series
  Car_D             150          Time Series
  ABUS_A            0            Time Series
  ABUS_C            0            Time Series
  green_A           150          Time Series
  green_B           150          Time Series
  green_C           150          Time Series
  green_D           150          Time Series
  dummy_green_go    150          Time Series
  dummy_green_left  150          Time Series


********************************************************************************
*                                  Processing                                  *
********************************************************************************

                                    Process                             Routing

 Entity   Location     Operation            Blk  Output           Destination  Rule        Move Logic
 -------- ------------ ------------------   ---- ---------------- ------------ ----------  ------------
 green_A  intersection if num_abus>0 then
                       route 1
                       else
                          begin
                          inc num_green_A
                          route 2
```

```
                         end

                                          1    dummy_green_left EXIT         FIRST 1
                                          2    green_A          pass         FIRST 1
green_A  pass         wait 16sec          1    green_A          EXIT         FIRST 1     dec num_green_A

green_B  intersection if num_abus>0 then
                      route 1
                      else
                         begin
                         inc num_green_B
                         route 2
                         end              1    dummy_green_left EXIT         FIRST 1
                                          2    green_B          pass         FIRST 1
green_B  pass         wait 10sec          1    green_B          EXIT         FIRST 1     dec num_green_B
green_C  intersection if num_abus>0 then
                      route 1
                      else
                        begin
                        inc num_green_C
                        route 2
                        end

                                          1    dummy_green_go   EXIT         FIRST 1
                                          2    green_C          pass         FIRST 1
green_C  pass         wait 16sec          1    green_C          EXIT         FIRST 1     dec num_green_C
green_D  intersection if num_abus>0 then
                      route 1
                      else
                         begin
                         inc num_green_D
                         route 2
                         end              1    dummy_green_go   EXIT         FIRST 1
                                          2    green_D          pass         FIRST 1
green_D  pass         wait 10sec          1    green_D          EXIT         FIRST 1     dec num_green_D
ABUS_A   intersection time1 = clock()     1    ABUS_A           Abus_lane_AC FIRST 1
ABUS_A   Abus_lane_AC
```

```
                                          1    ABUS_A           sensor_A     FIRST 1

ABUS_A   sensor_A     inc num_abus

                      if num_abus_C>0 then
                      wait until num_abus_C=0

                      inc num_abus_A
                                          1    ABUS_A           to_D,500     0.100000 1  time1 = clock() - time1
                                                                                         var1 = time1
                                                                                         move for 3sec

                                                                to_C,500     0.800000    time1 = clock() - time1
                                                                                         var2 = time1
                                                                                         move for 3sec


                                                                to_B,500     0.100000    time1 = clock() - time1
                                                                                         var3 = time1
                                                                                         move for 3sec

ABUS_A   to_D         inc num_abus_to_D
                      wait 5sec

                      dec num_abus_to_D
                      dec num_abus_A
                      dec num_abus        1    ABUS_A           EXIT         FIRST 1

ABUS_A   to_C         inc num_abus_to_C
                      wait 5sec

                      dec num_abus_to_C
                      dec num_abus_A
                      dec num_abus        1    ABUS_A           EXIT         FIRST 1
ABUS_A   to_B         inc num_abus_to_B
                      wait 5sec
```

```
                  dec num_abus_to_B
                  dec num_abus_A
                  dec num_abus         1    ABUS_A           EXIT         FIRST 1
Car_A    intersection                  1    Car_A            A1_car_input FIRST 1
Car_A    A1_car_input                  1    Car_A            A1_L_queue,1 0.100000 1
                                                             A1_G_queue,1 0.800000
                                                             A1_R_queue,1 0.100000
Car_A    A1_L_queue                    1    Car_A            A1_L_lane    FIRST 1



Car_A    A1_G_queue                    1    Car_A            A1_G_lane    FIRST 1
Car_A    A1_R_queue                    1    Car_A            A1_R_lane    FIRST 1
Car_A    A1_L_lane    time_a=clock()

                                       1    Car_A            to_D         FIRST 1     if num_green_A<1 then
                                                                                      wait until num_green_A=1

                                                                                      move for 2sec

                                                                                      time_a=clock()-time_a
                                                                                      var=time_a

Car_A    A1_G_lane
                                       1    Car_A            to_C         FIRST 1     if num_green_A<1 then
                                                                                      wait until num_green_A=1
                                                                                      else
                                                                                        begin
                                                                                        if num_abus_to_B>0 then
                                                                                        wait until num_abus_to_B=0
                                                                                        end
                                                                                      move for 2sec
Car_A    A1_R_lane                     1    Car_A            to_B         FIRST 1     if num_abus_to_B>0 then
                                                                                      wait until num_abus_to_B=0
                                                                                      move for 2sec



Car_B    intersection                  1    Car_B            B1_car_input FIRST 1
Car_B    B1_car_input                  1    Car_B            B1_L_queue   0.400000 1
                                                             B1_G_queue   0.200000
```

```
                                                      B1_R_queue   0.400000
Car_B    B1_L_queue                      1    Car_B       B1_L_lane    FIRST 1
Car_B    B1_G_queue                      1    Car_B       B1_G_lane    FIRST 1
Car_B    B1_R_queue                      1    Car_B       B1_R_lane    FIRST 1
Car_B    B1_L_lane                       1    Car_B       to_A         FIRST 1     if num_green_B<1 then
                                                                                   wait until num_green_B>0
                                                                                   move for 2sec

Car_B    B1_G_lane                       1    Car_B       to_D         FIRST 1     if num_green_B<1 then
                                                                                   wait until num_green_B>0
                                                                                   move for 2sec


Car_B    B1_R_lane                       1    Car_B       to_C         FIRST 1     move for 2sec
Car_C    intersection                    1    Car_C       C1_car_input FIRST 1
Car_C    C1_car_input                    1    Car_C       C1_L_queue   0.100000 1
                                                          C1_G_queue   0.800000
                                                          C1_R_queue   0.100000
Car_C    C1_L_queue                      1    Car_C       C1_L_lane    FIRST 1
Car_C    C1_G_queue                      1    Car_C       C1_G_lane    FIRST 1
Car_C    C1_R_queue                      1    Car_C       C1_R_lane    FIRST 1
Car_C    C1_L_lane                       1    Car_C       to_B         FIRST 1     if num_green_C<1 then
                                                                                   wait until num_green_C>0
                                                                                   move for 2sec


Car_C    C1_G_lane                       1    Car_C       to_A         FIRST 1     if num_green_C<1 then
                                                                                   wait until num_green_C=1
                                                                                   else
                                                                                     begin
                                                                                     if num_abus_to_D>0 then
                                                                                     wait until num_abus_to_D=0
                                                                                     end
                                                                                   move for 2sec

Car_C    C1_R_lane                       1    Car_C       to_D         FIRST 1     if num_abus_to_D>0 then
                                                                                   wait until num_abus_to_D=0
                                                                                   move for 2sec


Car_D    intersection                    1    Car_D       D1_car_input FIRST 1
Car_D    D1_car_input                    1    Car_D       D1_L_queue   0.400000 1
                                                          D1_G_queue   0.200000
                                                          D1_R_queue   0.400000
Car_D    D1_L_queue                      1    Car_D       D1_L_lane    FIRST 1
Car_D    D1_G_queue                      1    Car_D       D1_G_lane    FIRST 1
Car_D    D1_R_queue                      1    Car_D       D1_R_lane    FIRST 1
```

```
Car_D    D1_L_lane                          1    Car_D            to_C        FIRST 1     if num_green_D<1 then
                                                                                          wait until num_green_D>0
                                                                                          move for 2sec


Car_D    D1_G_lane                          1    Car_D            to_B        FIRST 1     if num_green_D<1 then
                                                                                          wait until num_green_D>0
                                                                                          move for 2sec

Car_D    D1_R_lane                          1    Car_D            to_A        FIRST 1     move for 2sec
ALL      to_A                               1    ALL              EXIT        FIRST 1
ALL      to_B                               1    ALL              EXIT        FIRST 1
ALL      to_C                               1    ALL              EXIT        FIRST 1
ALL      to_D                               1    ALL              EXIT        FIRST 1
ABUS_C   intersection                       1    ABUS_C           abus_lane_CA FIRST 1
ABUS_C   abus_lane_CA


                                            1    ABUS_C           sensor_C    FIRST 1
ABUS_C   sensor_C     inc num_abus

                      if num_abus_A>0 then
                      wait until num_abus_A=0

                      inc num_abus_C

                                            1    ABUS_C           to_B        0.250000 1  move for 3sec
                                                                  to_A        0.500000    move for 3sec
                                                                  to_D        0.250000    move for 3sec
ABUS_C   to_B         inc num_abus_to_B
                      wait 5sec

                      dec num_abus_to_B
                      dec num_abus_C
                      dec num_abus
                                            1    ABUS_C           EXIT        FIRST 1
ABUS_C   to_A         inc num_abus_to_A
                      wait 5sec
```

```
                      dec num_abus_to_A
                      dec num_abus_C
                      dec num_abus         1    ABUS_C           EXIT         FIRST 1
 ABUS_C   to_D        inc num_abus_to_D
                      wait 5sec

                      dec num_abus_to_D
                      dec num_abus_C
                      dec num_abus         1    ABUS_C           EXIT         FIRST 1

********************************************************************************
*                                   Arrivals                                   *
********************************************************************************

  Entity   Location     Qty each   First Time Occurrences Frequency  Logic
  -------- ------------ ---------- ---------- ----------- ---------- ------------
  Car_A    intersection 1          0          inf         E(2)sec
  Car_B    intersection 1          0          INF         E(2)sec
  Car_C    intersection 1          0          INF         E(2)sec
  Car_D    intersection 1          0          INF         E(2)sec
  ABUS_A   intersection 1          0          INF         N(10,1)min

  ABUS_C   intersection 1          2.5 min    INF         N(10,1)min
  green_A  intersection 1          0sec       INF         60sec


  green_B  intersection 1          18sec      INF         60sec
  green_C  intersection 1          30sec      INF         60sec




  green_D  intersection 1          48sec      INF         60sec

********************************************************************************
*                                  Attributes                                  *
********************************************************************************

  ID             Type         Classification
  -------------- ------------ --------------
  time1          Real         Entity
  time2          Real         Entity
  Time3          Real         Entity
  num_abus_left  Integer      Entity
```

```
  num_abus_go    Integer      Entity
  num_abus_right Integer      Entity
  time_a         Real         Entity


********************************************************************************
*                              Variables (global)                              *
********************************************************************************

  ID            Type         Initial value Stats
  ------------- ------------ ------------- -----------
  num_abus      Integer      0             Time Series
  num_abus_A    Integer      0             Time Series
  num_abus_C    Integer      0             Time Series
  num_abus_to_D Integer      0             Time Series
  num_abus_to_C Integer      0             Time Series
  num_abus_to_B Integer      0             Time Series
  num_abus_to_A Integer      0             Time Series
  num_green_A   Integer      0             Time Series
  num_green_B   Integer      0             Time Series
  num_green_C   Integer      0             Time Series
  num_green_D   Integer      0             Time Series
  var1          Real         0             Time Series
  var2          Real         0             Time Series
  var3          Real         0             Time Series
  var           Real         0             Time Series
```

Appendix C: The Structure of Model 5 – Simulation of a Four-intersection Corridor

```
********************************************************************************
*                                                                              *
*                         Formatted Listing of Model 5:                        *
*                 C:\ProMod4\models\masterproj\proj1-merge-e.MOD               *
*                                                                              *
********************************************************************************

  Time Units:                        Minutes
  Distance Units:                    Feet


********************************************************************************
*                                  Locations                                   *
********************************************************************************

  Name              Cap Units Stats       Rules               Cost
  ----------------- --- ----- ----------- ------------------- ------------
  inters_int1       inf 1     Time Series Oldest, ,
  start_A_int1      inf 1     Time Series Oldest, ,
  receive_A_int1    inf 1     Time Series Oldest, ,
  A_L_queue_int1    inf 1     Time Series Oldest, ,
  A_G_queue_int1    inf 1     Time Series Oldest, ,
  A_R_queue_int1    inf 1     Time Series Oldest, ,
  A_L_lane_int1     5   1     Time Series Oldest, FIFO,
  A_G_lane_int1     5   1     Time Series Oldest, FIFO,
  A_R_lane_int1     5   1     Time Series Oldest, FIFO,
  start_B_int1      inf 1     Time Series Oldest, ,
  receive_B_int1    inf 1     Time Series Oldest, ,
  B_L_queue_int1    inf 1     Time Series Oldest, ,
  B_G_queue_int1    inf 1     Time Series Oldest, ,
  B_R_queue_int1    inf 1     Time Series Oldest, ,
  B_L_lane_int1     5   1     Time Series Oldest, FIFO, First
  B_G_lane_int1     5   1     Time Series Oldest, FIFO,
  B_R_lane_int1     5   1     Time Series Oldest, FIFO,
  start_C_int1      inf 1     Time Series Oldest, ,
  receive_C_int1    inf 1     Time Series Oldest, ,
  C_L_queue_int1    inf 1     Time Series Oldest, ,
  C_G_queue_int1    inf 1     Time Series Oldest, ,
  C_R_queue_int1    inf 1     Time Series Oldest, ,
  C_L_lane_int1     5   1     Time Series Oldest, FIFO,
  C_G_lane_int1     5   1     Time Series Oldest, FIFO,
  C_R_lane_int1     5   1     Time Series Oldest, FIFO,
  start_D_int1      inf 1     Time Series Oldest, ,
```

```
receive_D_int1    inf 1     Time Series Oldest, ,
D_L_queue_int1    inf 1     Time Series Oldest, ,
D_G_queue_int1    inf 1     Time Series Oldest, ,
D_R_queue_int1    inf 1     Time Series Oldest, ,
D_L_lane_int1     5   1     Time Series Oldest, FIFO,
D_G_lane_int1     5   1     Time Series Oldest, FIFO,
D_R_lane_int1     5   1     Time Series Oldest, FIFO,
Abus_lane_AC_int1 inf 1     Time Series Oldest, FIFO,
abus_lane_CA_int1 inf 1     Time Series Oldest, FIFO,
left_int1         inf 1     Time Series Oldest, ,
go_int1           inf 1     Time Series Oldest, ,
right_int1        inf 1     Time Series Oldest, ,
sensor_A_int1     1   1     Time Series Oldest, ,
sensor_C_int1     1   1     Time Series Oldest, ,
inters_int2       inf 1     Time Series Oldest, ,
start_A_int2      inf 1     Time Series Oldest, ,
receive_A_int2    inf 1     Time Series Oldest, ,
A_L_queue_int2    inf 1     Time Series Oldest, ,
A_G_queue_int2    inf 1     Time Series Oldest, ,
A_R_queue_int2    inf 1     Time Series Oldest, ,
A_L_lane_int2     5   1     Time Series Oldest, FIFO,
A_G_lane_int2     5   1     Time Series Oldest, FIFO,
A_R_lane_int2     5   1     Time Series Oldest, FIFO,
start_B_int2      inf 1     Time Series Oldest, ,
receive_B_int2    inf 1     Time Series Oldest, ,
B_L_queue_int2    inf 1     Time Series Oldest, ,
B_G_queue_int2    inf 1     Time Series Oldest, ,
B_R_queue_int2    inf 1     Time Series Oldest, ,
B_L_lane_int2     5   1     Time Series Oldest, FIFO, First
B_G_lane_int2     5   1     Time Series Oldest, FIFO,
B_R_lane_int2     5   1     Time Series Oldest, FIFO,
start_C_int2      inf 1     Time Series Oldest, ,
receive_C_int2    inf 1     Time Series Oldest, ,
C_L_queue_int2    inf 1     Time Series Oldest, ,
C_G_queue_int2    inf 1     Time Series Oldest, ,
C_R_queue_int2    inf 1     Time Series Oldest, ,
C_L_lane_int2     5   1     Time Series Oldest, FIFO,
C_G_lane_int2     5   1     Time Series Oldest, FIFO,
C_R_lane_int2     5   1     Time Series Oldest, FIFO,
start_D_int2      inf 1     Time Series Oldest, ,
receive_D_int2    inf 1     Time Series Oldest, ,
D_L_queue_int2    inf 1     Time Series Oldest, ,
D_G_queue_int2    inf 1     Time Series Oldest, ,
D_R_queue_int2    inf 1     Time Series Oldest, ,
D_L_lane_int2     5   1     Time Series Oldest, FIFO,
D_G_lane_int2     5   1     Time Series Oldest, FIFO,
D_R_lane_int2     5   1     Time Series Oldest, FIFO,
```

```
Abus_lane_AC_int2 inf 1     Time Series Oldest, FIFO,
abus_lane_CA_int2 inf 1     Time Series Oldest, FIFO,
left_int2         inf 1     Time Series Oldest, ,
go_int2           inf 1     Time Series Oldest, ,
right_int2        inf 1     Time Series Oldest, ,
sensor_A_int2     1   1     Time Series Oldest, ,
sensor_C_int2     1   1     Time Series Oldest, ,
inters_int3       inf 1     Time Series Oldest, ,
start_A_int3      inf 1     Time Series Oldest, ,
receive_A_int3    inf 1     Time Series Oldest, ,
A_L_queue_int3    inf 1     Time Series Oldest, ,
A_G_queue_int3    inf 1     Time Series Oldest, ,
A_R_queue_int3    inf 1     Time Series Oldest, ,
A_L_lane_int3     5   1     Time Series Oldest, FIFO,
A_G_lane_int3     5   1     Time Series Oldest, FIFO,
A_R_lane_int3     5   1     Time Series Oldest, FIFO,
start_B_int3      inf 1     Time Series Oldest, ,
receive_B_int3    inf 1     Time Series Oldest, ,
B_L_queue_int3    inf 1     Time Series Oldest, ,
B_G_queue_int3    inf 1     Time Series Oldest, ,
B_R_queue_int3    inf 1     Time Series Oldest, ,
B_L_lane_int3     5   1     Time Series Oldest, FIFO, First
B_G_lane_int3     5   1     Time Series Oldest, FIFO,
B_R_lane_int3     5   1     Time Series Oldest, FIFO,
start_C_int3      inf 1     Time Series Oldest, ,
receive_C_int3    inf 1     Time Series Oldest, ,
C_L_queue_int3    inf 1     Time Series Oldest, ,
C_G_queue_int3    inf 1     Time Series Oldest, ,
C_R_queue_int3    inf 1     Time Series Oldest, ,
C_L_lane_int3     5   1     Time Series Oldest, FIFO,
C_G_lane_int3     5   1     Time Series Oldest, FIFO,
C_R_lane_int3     5   1     Time Series Oldest, FIFO,
start_D_int3      inf 1     Time Series Oldest, ,
receive_D_int3    inf 1     Time Series Oldest, ,
D_L_queue_int3    inf 1     Time Series Oldest, ,
D_G_queue_int3    inf 1     Time Series Oldest, ,
D_R_queue_int3    inf 1     Time Series Oldest, ,
D_L_lane_int3     5   1     Time Series Oldest, FIFO,
D_G_lane_int3     5   1     Time Series Oldest, FIFO,
D_R_lane_int3     5   1     Time Series Oldest, FIFO,
Abus_lane_AC_int3 inf 1     Time Series Oldest, FIFO,
abus_lane_CA_int3 inf 1     Time Series Oldest, FIFO,
left_int3         inf 1     Time Series Oldest, ,
go_int3           inf 1     Time Series Oldest, ,
right_int3        inf 1     Time Series Oldest, ,
sensor_A_int3     1   1     Time Series Oldest, ,
sensor_C_int3     1   1     Time Series Oldest, ,
```

```
inters_int4       inf 1     Time Series Oldest, ,
start_A_int4      inf 1     Time Series Oldest, ,
receive_A_int4    inf 1     Time Series Oldest, ,
A_L_queue_int4    inf 1     Time Series Oldest, ,
A_G_queue_int4    inf 1     Time Series Oldest, ,
A_R_queue_int4    inf 1     Time Series Oldest, ,
A_L_lane_int4     5   1     Time Series Oldest, FIFO,
A_G_lane_int4     5   1     Time Series Oldest, FIFO,
A_R_lane_int4     5   1     Time Series Oldest, FIFO,
start_B_int4      inf 1     Time Series Oldest, ,
receive_B_int4    inf 1     Time Series Oldest, ,
B_L_queue_int4    inf 1     Time Series Oldest, ,
B_G_queue_int4    inf 1     Time Series Oldest, ,
B_R_queue_int4    inf 1     Time Series Oldest, ,
B_L_lane_int4     5   1     Time Series Oldest, FIFO, First
B_G_lane_int4     5   1     Time Series Oldest, FIFO,
B_R_lane_int4     5   1     Time Series Oldest, FIFO,
start_C_int4      inf 1     Time Series Oldest, ,
receive_C_int4    inf 1     Time Series Oldest, ,
C_L_queue_int4    inf 1     Time Series Oldest, ,
C_G_queue_int4    inf 1     Time Series Oldest, ,
C_R_queue_int4    inf 1     Time Series Oldest, ,
C_L_lane_int4     5   1     Time Series Oldest, FIFO,
C_G_lane_int4     5   1     Time Series Oldest, FIFO,
C_R_lane_int4     5   1     Time Series Oldest, FIFO,
start_D_int4      inf 1     Time Series Oldest, ,
receive_D_int4    inf 1     Time Series Oldest, ,
D_L_queue_int4    inf 1     Time Series Oldest, ,
D_G_queue_int4    inf 1     Time Series Oldest, ,
D_R_queue_int4    inf 1     Time Series Oldest, ,
D_L_lane_int4     5   1     Time Series Oldest, FIFO,
D_G_lane_int4     5   1     Time Series Oldest, FIFO,
D_R_lane_int4     5   1     Time Series Oldest, FIFO,
Abus_lane_AC_int4 inf 1     Time Series Oldest, FIFO,
abus_lane_CA_int4 inf 1     Time Series Oldest, FIFO,
left_int4         inf 1     Time Series Oldest, ,
go_int4           inf 1     Time Series Oldest, ,
right_int4        inf 1     Time Series Oldest, ,
sensor_A_int4     1   1     Time Series Oldest, ,
sensor_C_int4     1   1     Time Series Oldest, ,


********************************************************************************
*                                   Entities                                   *
********************************************************************************

Name              Speed (fpm)  Stats       Cost
```

```
---------------- ------------ ----------- ------------
Car              150          Time Series
ABUS_A           0            Time Series
ABUS_C           0            Time Series
green_left_AC1   150          Time Series
green_left_BD1   150          Time Series
green_go_AC1     150          Time Series
green_go_BD1     150          Time Series
green_left_AC2   150          Time Series
green_left_BD2   150          Time Series
green_go_AC2     150          Time Series
green_go_BD2     150          Time Series
green_left_AC3   150          Time Series
green_left_BD3   150          Time Series
green_go_AC3     150          Time Series
green_go_BD3     150          Time Series
green_left_AC4   150          Time Series
green_left_BD4   150          Time Series
green_go_AC4     150          Time Series
green_go_BD4     150          Time Series
dummy_green_go   150          Time Series
dummy_green_left 150          Time Series
```

```
*********************************************************************************
*                                  Processing                                   *
*********************************************************************************

                                            Process                                    Routing

 Entity         Location          Operation            Blk  Output           Destination        Rule        Move Logic
 -------------- ----------------- ------------------   ---- ---------------- ------------------ ----------  ------------
 green_go_AC1   inters_int1       if num_abus_int1>0 then
                                  route 1
                                  else
                                    begin
                                    inc num_green_go_AC_int1
                                    route 2
                                    end



                                                       1    dummy_green_go   EXIT               FIRST 1
                                                       2    green_go_AC1     go_int1            FIRST 1
```

```
 green_go_AC1   go_int1           wait 16sec           1    green_go_AC1     EXIT               FIRST 1     dec num_green_go_AC_int1
 green_left_AC1 inters_int1       if num_abus_int1>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_AC_int1
                                     route 2
                                     end


                                                       1    dummy_green_left EXIT               FIRST 1
                                                       2    green_left_AC1   left_int1          FIRST 1
 green_left_AC1 left_int1         wait 16sec           1    green_left_AC1   EXIT               FIRST 1     dec
num_green_left_AC_int1
 green_go_BD1   inters_int1       if num_abus_int1>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_go_BD_int1
                                     route 2
                                     end               1    dummy_green_go   EXIT               FIRST 1
                                                       2    green_go_BD1     go_int1            FIRST 1
 green_go_BD1   go_int1           wait 16sec           1    green_go_BD1     EXIT               FIRST 1     dec num_green_go_BD_int1

 green_left_BD1 inters_int1       if num_abus_int1>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_BD_int1
                                     route 2
                                     end               1    dummy_green_left EXIT               FIRST 1
                                                       2    green_left_BD1   left_int1          FIRST 1
 green_left_BD1 left_int1         wait 16sec           1    green_left_BD1   EXIT               FIRST 1     dec
num_green_left_BD_int1
 ABUS_A         inters_int1       time1 = clock()      1    ABUS_A           Abus_lane_AC_int1  FIRST 1
 ABUS_A         Abus_lane_AC_int1


                                                       1    ABUS_A           sensor_A_int1      FIRST 1


 ABUS_A         sensor_A_int1     inc num_abus_int1
                                  inc num_abus_A_int1

                                                       1    ABUS_A           receive_C_int1,500 FIRST 1     time1 = clock() - time1
```

```
                                                                                                                              var1 = time1
                                                                                                                              move for 3sec

 ABUS_A         receive_C_int1    inc num_abus_to_C_int1
                                  wait 5sec

                                  dec num_abus_to_C_int1
                                  dec num_abus_A_int1
                                  dec num_abus_int1
                                                       1    ABUS_A          inters_int2         FIRST 1     move for N(5,0.5)min
 ABUS_C         inters_int1       time1 = clock()      1    ABUS_C          abus_lane_CA_int1   FIRST 1
 ABUS_C         abus_lane_CA_int1



                                                       1    ABUS_C          sensor_C_int1       FIRST 1


 ABUS_C         sensor_C_int1     inc num_abus_int1
                                  inc num_abus_C_int1
                                                       1    ABUS_C          receive_A_int1      FIRST 1     move for 3sec
 ABUS_C         receive_A_int1    inc num_abus_to_A_int1
                                  wait 5sec

                                  dec num_abus_to_A_int1
                                  dec num_abus_C_int1
                                  dec num_abus_int1
                                                       1    ABUS_C          EXIT                FIRST 1
 Car            start_A_int1                           1    Car             A_L_queue_int1,1    0.250000 1
                                                                            A_G_queue_int1,1    0.500000
                                                                            A_R_queue_int1,1    0.250000
 Car            A_L_queue_int1                         1    Car             A_L_lane_int1       FIRST 1



 Car            A_G_queue_int1                         1    Car             A_G_lane_int1       FIRST 1
 Car            A_R_queue_int1                         1    Car             A_R_lane_int1       FIRST 1
 Car            A_L_lane_int1     time_a=clock()

                                                       1    Car             receive_D_int1      FIRST 1     if
num_green_left_AC_int1<1 then
                                                                                                            wait until
num_green_left_AC_int1=1

                                                                                                            move for 2sec
```

```
                                                                                      time_a=clock()-time_a
                                                                                      var=time_a

 Car            A_G_lane_int1
                                                  1    Car             receive_C_int1     FIRST 1     if num_green_go_AC_int1<
1 then
                                                                                      wait until
num_green_go_AC_int1=1

                                                                                      else
                                                                                        begin
                                                                                          if
num_abus_to_B_int1>0 then
                                                                                          wait until
num_abus_to_B_int1=0
                                                                                        end

                                                                                      move for 2sec
 Car            A_R_lane_int1                     1    Car             receive_B_int1     FIRST 1     if num_abus_to_B_int1>0
then

                                                                                      wait until
num_abus_to_B_int1=0
                                                                                      move for 2sec



 Car            start_B_int1                      1    Car             B_L_queue_int1     0.250000 1
                                                                       B_G_queue_int1     0.500000
                                                                       B_R_queue_int1     0.250000
 Car            B_L_queue_int1                    1    Car             B_L_lane_int1      FIRST 1
 Car            B_G_queue_int1                    1    Car             B_G_lane_int1      FIRST 1
 Car            B_R_queue_int1                    1    Car             B_R_lane_int1      FIRST 1
 Car            B_L_lane_int1                     1    Car             receive_A_int1     FIRST 1     if
num_green_left_BD_int1<1 then
                                                                                      wait until
num_green_left_BD_int1=1


                                                                                      move for 2sec

 Car            B_G_lane_int1                     1    Car             receive_D_int1     FIRST 1     if num_green_go_BD_int1<1
then
                                                                                      wait until
num_green_go_BD_int1=1


                                                                                      move for 2sec
```

```
 Car            B_R_lane_int1                         1    Car              receive_C_int1     FIRST 1     move for 2sec
 Car            start_C_int1                          1    Car              C_L_queue_int1     0.250000 1
                                                                            C_G_queue_int1     0.500000
                                                                            C_R_queue_int1     0.250000
 Car            C_L_queue_int1                        1    Car              C_L_lane_int1      FIRST 1
 Car            C_G_queue_int1                        1    Car              C_G_lane_int1      FIRST 1
 Car            C_R_queue_int1                        1    Car              C_R_lane_int1      FIRST 1
 Car            C_L_lane_int1                         1    Car              receive_B_int1     FIRST 1     if
num_green_left_AC_int1<1 then
                                                                                                           wait until
num_green_left_AC_int1=1

                                                                                                           move for 2sec


 Car            C_G_lane_int1                         1    Car              receive_A_int1     FIRST 1     if num_green_go_AC_int1<
1 then
                                                                                                           wait until
num_green_go_AC_int1=1
                                                                                                           else
                                                                                                             begin
                                                                                                               if
num_abus_to_D_int1>0 then
                                                                                                               wait until
num_abus_to_D_int1=0
                                                                                                             end

                                                                                                           move for 2sec

 Car            C_R_lane_int1                         1    Car              receive_D_int1     FIRST 1     if num_abus_to_D_int1>0
then

                                                                                                           wait until
num_abus_to_D_int1=0


                                                                                                           move for 2sec


 Car            start_D_int1                          1    Car              D_L_queue_int1     0.250000 1
                                                                            D_G_queue_int1     0.500000
                                                                            D_R_queue_int1     0.250000
 Car            D_L_queue_int1                        1    Car              D_L_lane_int1      FIRST 1
 Car            D_G_queue_int1                        1    Car              D_G_lane_int1      FIRST 1
 Car            D_R_queue_int1                        1    Car              D_R_lane_int1      FIRST 1
 Car            D_L_lane_int1                         1    Car              receive_C_int1     FIRST 1     if
num_green_left_BD_int1<1 then
```

```
                                                                                                                              wait until
num_green_left_BD_int1=1
                                                                                                                              move for 2sec

 Car             D_G_lane_int1                           1    Car               receive_B_int1     FIRST 1     if num_green_go_BD_int1<1
then
                                                                                                                               wait until
num_green_go_BD_int1=1
                                                                                                                              move for 2sec
 Car             D_R_lane_int1                           1    Car               receive_A_int1     FIRST 1     move for 2sec
 Car             receive_A_int1    if car_ID=0 then
                                   begin
                                    road_time_0=clock()-road_time_0
                                    total_road_time_0=total_road_time_0+road_time_0
                                    inc num_through_0
                                    avg_road_time_0=total_road_time_0/num_through_0
                                   end
                                                         1    Car               EXIT               FIRST 1
 Car             receive_B_int1    if car_ID=1 then
                                   begin
                                    inc num_cross_int1
                                    road_time_1=clock()-road_time_1
                                    total_road_time_int1=total_road_time_int1+road_time_1
                                    avg_road_time_int1=total_road_time_int1/num_cross_int1
                                   end

                                                         1    Car               EXIT               FIRST 1
 Car             receive_C_int1                          1    Car               start_A_int2       FIRST 1     move for N(5,0.5)min
 Car             receive_D_int1    if car_ID=1 then
                                   begin
                                    inc num_cross_int1
                                    road_time_1=clock()-road_time_1
                                    total_road_time_int1=total_road_time_int1+road_time_1
                                    avg_road_time_int1=total_road_time_int1/num_cross_int1
                                   end

                                                         1    Car               EXIT               FIRST 1
 green_go_AC2    inters_int2       if num_abus_int2>0 then
                                   route 1
                                   else
                                     begin
```

```
                                   inc num_green_go_AC_int2
                                   route 2
                                   end


                                                     1    dummy_green_go   EXIT              FIRST 1
                                                     2    green_go_AC2     go_int2           FIRST 1
 green_go_AC2   go_int2          wait 16sec          1    green_go_AC2     EXIT              FIRST 1     dec num_green_go_AC_int2
 green_left_AC2 inters_int2      if num_abus_int2>0 then
                                 route 1
                                 else
                                    begin
                                    inc num_green_left_AC_int2
                                    route 2
                                    end


                                                     1    dummy_green_left EXIT              FIRST 1
                                                     2    green_left_AC2   left_int2         FIRST 1
 green_left_AC2 left_int2        wait 16sec          1    green_left_AC2   EXIT              FIRST 1     dec
num_green_left_AC_int2
 green_go_BD2   inters_int2      if num_abus_int2>0 then
                                 route 1
                                 else
                                    begin
                                    inc num_green_go_BD_int2
                                    route 2
                                    end              1    dummy_green_go   EXIT              FIRST 1
                                                     2    green_go_BD2     go_int2           FIRST 1
 green_go_BD2   go_int2          wait 10sec          1    green_go_BD2     EXIT              FIRST 1     dec num_green_go_BD_int2

 green_left_BD2 inters_int2      if num_abus_int2>0 then
                                 route 1
                                 else
                                    begin
                                    inc num_green_left_BD_int2
                                    route 2
                                    end              1    dummy_green_left EXIT              FIRST 1
                                                     2    green_left_BD2   left_int2         FIRST 1
 green_left_BD2 left_int2        wait 10sec          1    green_left_BD2   EXIT              FIRST 1     dec
num_green_left_BD_int2
 ABUS_A         inters_int2      time1 = clock()     1    ABUS_A           Abus_lane_AC_int2 FIRST 1
 ABUS_A         Abus_lane_AC_int2
```

			1	ABUS_A	sensor_A_int2	FIRST 1	
ABUS_A	sensor_A_int2	inc num_abus_int2 inc num_abus_A_int2					
			1	ABUS_A	receive_C_int2,500	FIRST 1	time1 = clock() - time1 var1int2 = time1 move for 3sec
ABUS_A	receive_C_int2	inc num_abus_to_C_int2 wait 5sec dec num_abus_to_C_int2 dec num_abus_A_int2 dec num_abus_int2					
			1	ABUS_A	inters_int3	FIRST 1	move for N(4.5,0.45)min
ABUS_C	inters_int2	time1 = clock()	1	ABUS_C	abus_lane_CA_int2	FIRST 1	
ABUS_C	abus_lane_CA_int2						
			1	ABUS_C	sensor_C_int2	FIRST 1	
ABUS_C	sensor_C_int2	inc num_abus_int2 inc num_abus_C_int2					
			1	ABUS_C	receive_A_int2	FIRST 1	move for 3sec
ABUS_C	receive_A_int2	inc num_abus_to_A_int2 wait 5sec dec num_abus_to_A_int2 dec num_abus_C_int2 dec num_abus_int2					
			1	ABUS_C	inters_int1	FIRST 1	move for N(5,0.5)min
Car	start_A_int2		1	Car	A_L_queue_int2,1 A_G_queue_int2,1 A_R_queue_int2,1	0.100000 1 0.800000 0.100000	
Car	A_L_queue_int2		1	Car	A_L_lane_int2	FIRST 1	
Car	A_G_queue_int2		1	Car	A_G_lane_int2	FIRST 1	
Car	A_R_queue_int2		1	Car	A_R_lane_int2	FIRST 1	

```
 Car            A_L_lane_int2     time_a=clock()
                                                     1    Car               receive_D_int2     FIRST 1     if
num_green_left_AC_int2<1 then
                                                                                                           wait until
num_green_left_AC_int2=1

                                                                                                           move for 2sec

                                                                                                           time_a=clock()-time_a
                                                                                                           varint2=time_a

 Car            A_G_lane_int2
                                                     1    Car               receive_C_int2     FIRST 1     if num_green_go_AC_int2<
1 then
                                                                                                           wait until
num_green_go_AC_int2=1

                                                                                                           else
                                                                                                             begin
                                                                                                               if
num_abus_to_B_int2>0 then
                                                                                                               wait until
num_abus_to_B_int2=0
                                                                                                             end

                                                                                                           move for 2sec
 Car            A_R_lane_int2                        1    Car               receive_B_int2     FIRST 1     if num_abus_to_B_int2>0
then

                                                                                                           wait until
num_abus_to_B_int2=0
                                                                                                           move for 2sec




 Car            start_B_int2                         1    Car               B_L_queue_int2     0.400000 1
                                                                            B_G_queue_int2     0.200000
                                                                            B_R_queue_int2     0.400000
 Car            B_L_queue_int2                       1    Car               B_L_lane_int2      FIRST 1
 Car            B_G_queue_int2                       1    Car               B_G_lane_int2      FIRST 1
 Car            B_R_queue_int2                       1    Car               B_R_lane_int2      FIRST 1
 Car            B_L_lane_int2                        1    Car               receive_A_int2     FIRST 1     if
num_green_left_BD_int2<1 then
```

```
                                                                                                          wait until
num_green_left_BD_int2=1

                                                                                                          move for 2sec

 Car            B_G_lane_int2                         1    Car              receive_D_int2     FIRST 1     if num_green_go_BD_int2<1
then
                                                                                                          wait until
num_green_go_BD_int2=1

                                                                                                          move for 2sec


 Car            B_R_lane_int2                         1    Car              receive_C_int2     FIRST 1     move for 2sec
 Car            start_C_int2                          1    Car              C_L_queue_int2     0.100000 1
                                                                            C_G_queue_int2     0.800000
                                                                            C_R_queue_int2     0.100000
 Car            C_L_queue_int2                        1    Car              C_L_lane_int2      FIRST 1
 Car            C_G_queue_int2                        1    Car              C_G_lane_int2      FIRST 1
 Car            C_R_queue_int2                        1    Car              C_R_lane_int2      FIRST 1
 Car            C_L_lane_int2                         1    Car              receive_B_int2     FIRST 1     if
num_green_left_AC_int2<1 then
                                                                                                          wait until
num_green_left_AC_int2=1

                                                                                                          move for 2sec


 Car            C_G_lane_int2                         1    Car              receive_A_int2     FIRST 1     if num_green_go_AC_int2<
1 then
                                                                                                          wait until
num_green_go_AC_int2=1
                                                                                                          else
                                                                                                            begin
                                                                                                              if
num_abus_to_D_int2>0 then
                                                                                                              wait until
num_abus_to_D_int2=0
                                                                                                            end

                                                                                                          move for 2sec

 Car            C_R_lane_int2                         1    Car              receive_D_int2     FIRST 1     if num_abus_to_D_int2>0
then

                                                                                                          wait until
num_abus_to_D_int2=0
```

```
                                                                                                                          move for 2sec


 Car            start_D_int2                                1    Car                D_L_queue_int2     0.400000 1
                                                                                    D_G_queue_int2     0.200000
                                                                                    D_R_queue_int2     0.400000
 Car            D_L_queue_int2                              1    Car                D_L_lane_int2      FIRST 1
 Car            D_G_queue_int2                              1    Car                D_G_lane_int2      FIRST 1
 Car            D_R_queue_int2                              1    Car                D_R_lane_int2      FIRST 1
 Car            D_L_lane_int2                               1    Car                receive_C_int2     FIRST 1     if
num_green_left_BD_int2<1 then
                                                                                                                   wait until
num_green_left_BD_int2=1

                                                                                                                   move for 2sec



 Car            D_G_lane_int2                               1    Car                receive_B_int2     FIRST 1     if num_green_go_BD_int2<1
then
                                                                                                                    wait until
num_green_go_BD_int2=1

                                                                                                                   move for 2sec

 Car            D_R_lane_int2                               1    Car                receive_A_int2     FIRST 1     move for 2sec
 Car            receive_A_int2                              1    Car                start_C_int1       FIRST 1     move for N(5,0.5)
 Car            receive_B_int2    if car_ID=2 then
                                  begin
                                   inc num_cross_int2
                                   road_time_2=clock()-road_time_2
                                   total_road_time_int2=total_road_time_int2+road_time_2
                                   avg_road_time_int2=total_road_time_int2/num_cross_int2
                                  end


                                                            1    Car                EXIT               FIRST 1
 Car            receive_C_int2                              1    Car                start_A_int3       FIRST 1     move for N(4.5,0.45)
 Car            receive_D_int2    if car_ID=2 then
                                  begin
                                   inc num_cross_int2
                                   road_time_2=clock()-road_time_2
                                   total_road_time_int2=total_road_time_int2+road_time_2
                                   avg_road_time_int2=total_road_time_int2/num_cross_int2
                                  end
```

```
                                           1    Car               EXIT                FIRST 1
 green_go_AC3   inters_int3       if num_abus_int3>0 then
                                  route 1
                                  else
                                    begin
                                    inc num_green_go_AC_int3
                                    route 2
                                    end

                                           1    dummy_green_go    EXIT                FIRST 1
                                           2    green_go_AC3      go_int3             FIRST 1
 green_go_AC3   go_int3           wait 16sec           1    green_go_AC3      EXIT                FIRST 1     dec num_green_go_AC_int3
 green_left_AC3 inters_int3       if num_abus_int3>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_AC_int3
                                     route 2
                                     end



                                           1    dummy_green_left  EXIT                FIRST 1
                                           2    green_left_AC3    left_int3           FIRST 1
 green_left_AC3 left_int3         wait 16sec           1    green_left_AC3    EXIT                FIRST 1     dec
num_green_left_AC_int3
 green_go_BD3   inters_int3       if num_abus_int3>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_go_BD_int3
                                     route 2
                                     end               1    dummy_green_go    EXIT                FIRST 1
                                                       2    green_go_BD3      go_int3             FIRST 1
 green_go_BD3   go_int3           wait 10sec           1    green_go_BD3      EXIT                FIRST 1     dec num_green_go_BD_int3

 green_left_BD3 inters_int3       if num_abus_int3>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_BD_int3
                                     route 2
                                     end               1    dummy_green_left  EXIT                FIRST 1
                                                       2    green_left_BD3    left_int3           FIRST 1
 green_left_BD3 left_int3         wait 10sec           1    green_left_BD3    EXIT                FIRST 1     dec
num_green_left_BD_int3
```

ABUS_A	inters_int3	time1 = clock()	1	ABUS_A	Abus_lane_AC_int3	FIRST 1	
ABUS_A	Abus_lane_AC_int3		1	ABUS_A	sensor_A_int3	FIRST 1	
ABUS_A	sensor_A_int3	inc num_abus_int3 inc num_abus_A_int3	1	ABUS_A	receive_C_int3,500	FIRST 1	time1 = clock() - time1 var1int3 = time1 move for 3sec
ABUS_A	receive_C_int3	inc num_abus_to_C_int3 wait 5sec dec num_abus_to_C_int3 dec num_abus_A_int3 dec num_abus_int3	1	ABUS_A	inters_int4	FIRST 1	move for N(6,0.6)
ABUS_C	inters_int3	time1 = clock()	1	ABUS_C	abus_lane_CA_int3	FIRST 1	
ABUS_C	abus_lane_CA_int3		1	ABUS_C	sensor_C_int3	FIRST 1	
ABUS_C	sensor_C_int3	inc num_abus_int3 inc num_abus_C_int3	1	ABUS_C	receive_A_int3	FIRST 1	move for 3sec
ABUS_C	receive_A_int3	inc num_abus_to_A_int3 wait 5sec dec num_abus_to_A_int3 dec num_abus_C_int3 dec num_abus_int3	1	ABUS_C	inters_int2	FIRST 1	move for N(4.5,0.45)min
Car	start_A_int3		1	Car	A_L_queue_int3,1 A_G_queue_int3,1 A_R_queue_int3,1	0.100000 1 0.800000 0.100000	
Car	A_L_queue_int3		1	Car	A_L_lane_int3	FIRST 1	
Car	A_G_queue_int3		1	Car	A_G_lane_int3	FIRST 1	
Car	A_R_queue_int3		1	Car	A_R_lane_int3	FIRST 1	

```
 Car            A_L_lane_int3     time_a=clock()
                                                     1    Car               receive_D_int3     FIRST 1     if
num_green_left_AC_int3<1 then
                                                                                                            wait until
num_green_left_AC_int3=1

                                                                                                            move for 2sec

                                                                                                            time_a=clock()-time_a
                                                                                                            varint3=time_a

 Car            A_G_lane_int3
                                                     1    Car               receive_C_int3     FIRST 1     if num_green_go_AC_int3<
1 then
                                                                                                            wait until
num_green_go_AC_int3=1

                                                                                                            else
                                                                                                              begin
                                                                                                                if
num_abus_to_B_int3>0 then
                                                                                                                wait until
num_abus_to_B_int3=0
                                                                                                              end

                                                                                                            move for 2sec
 Car            A_R_lane_int3                        1    Car               receive_B_int3     FIRST 1     if num_abus_to_B_int3>0
then

                                                                                                            wait until
num_abus_to_B_int3=0
                                                                                                            move for 2sec


 Car            start_B_int3                         1    Car               B_L_queue_int3     0.400000 1
                                                                            B_G_queue_int3     0.200000
                                                                            B_R_queue_int3     0.400000
 Car            B_L_queue_int3                       1    Car               B_L_lane_int3      FIRST 1
 Car            B_G_queue_int3                       1    Car               B_G_lane_int3      FIRST 1
 Car            B_R_queue_int3                       1    Car               B_R_lane_int3      FIRST 1
 Car            B_L_lane_int3                        1    Car               receive_A_int3     FIRST 1     if
num_green_left_BD_int3<1 then
                                                                                                            wait until
num_green_left_BD_int3=1

                                                                                                            move for 2sec
```

```
 Car            B_G_lane_int3                         1    Car              receive_D_int3     FIRST 1     if num_green_go_BD_int3<1
then
                                                                                                           wait until
num_green_go_BD_int3=1

                                                                                                           move for 2sec


 Car            B_R_lane_int3                         1    Car              receive_C_int3     FIRST 1     move for 2sec
 Car            start_C_int3                          1    Car              C_L_queue_int3     0.100000 1
                                                                            C_G_queue_int3     0.800000
                                                                            C_R_queue_int3     0.100000
 Car            C_L_queue_int3                        1    Car              C_L_lane_int3      FIRST 1
 Car            C_G_queue_int3                        1    Car              C_G_lane_int3      FIRST 1
 Car            C_R_queue_int3                        1    Car              C_R_lane_int3      FIRST 1
 Car            C_L_lane_int3                         1    Car              receive_B_int3     FIRST 1     if
num_green_left_AC_int3<1 then
                                                                                                           wait until
num_green_left_AC_int3=1

                                                                                                           move for 2sec


 Car            C_G_lane_int3                         1    Car              receive_A_int3     FIRST 1     if num_green_go_AC_int3<
1 then
                                                                                                           wait until
num_green_go_AC_int3=1
                                                                                                           else
                                                                                                             begin
                                                                                                               if
num_abus_to_D_int3>0 then
                                                                                                               wait until
num_abus_to_D_int3=0
                                                                                                             end

                                                                                                           move for 2sec
 Car            C_R_lane_int3                         1    Car              receive_D_int3     FIRST 1     if num_abus_to_D_int3>0
then

                                                                                                           wait until
num_abus_to_D_int3=0

                                                                                                           move for 2sec


 Car            start_D_int3                          1    Car              D_L_queue_int3     0.400000 1
```

```
                                                                                             D_G_queue_int3     0.200000
                                                                                             D_R_queue_int3     0.400000
 Car            D_L_queue_int3                                       1    Car                D_L_lane_int3      FIRST 1
 Car            D_G_queue_int3                                       1    Car                D_G_lane_int3      FIRST 1
 Car            D_R_queue_int3                                       1    Car                D_R_lane_int3      FIRST 1
 Car            D_L_lane_int3                                        1    Car                receive_C_int3     FIRST 1     if
num_green_left_BD_int3<1 then
                                                                                                                            wait until
num_green_left_BD_int3=1

                                                                                                                            move for 2sec



 Car            D_G_lane_int3                                        1    Car                receive_B_int3     FIRST 1     if num_green_go_BD_int3<1
then
                                                                                                                             wait until
num_green_go_BD_int3=1

                                                                                                                            move for 2sec

 Car            D_R_lane_int3                                        1    Car                receive_A_int3     FIRST 1     move for 2sec
 Car            receive_A_int3                                       1    Car                start_C_int2       FIRST 1     move for N(4.5,0.45)min

 Car            receive_B_int3    if car_ID=3 then
                                  begin
                                   inc num_cross_int3
                                   road_time_3=clock()-road_time_3
                                   total_road_time_int3=total_road_time_int3+road_time_3
                                   avg_road_time_int3=total_road_time_int3/num_cross_int3
                                  end


                                                                     1    Car                EXIT               FIRST 1
 Car            receive_C_int3                                       1    Car                start_A_int4       FIRST 1     move for N(6,0.6)min
 Car            receive_D_int3    if car_ID=3 then
                                  begin
                                   inc num_cross_int3
                                   road_time_3=clock()-road_time_3
                                   total_road_time_int3=total_road_time_int3+road_time_3
                                   avg_road_time_int3=total_road_time_int3/num_cross_int3
                                  end                                1    Car                EXIT               FIRST 1
 green_go_AC4   inters_int4       if num_abus_int4>0 then
                                  route 1
                                  else
                                    begin
                                    inc num_green_go_AC_int4
```

```
                                    route 2
                                    end



                                                       1    dummy_green_go   EXIT              FIRST 1
                                                       2    green_go_AC4     go_int4           FIRST 1
 green_go_AC4   go_int4           wait 16sec           1    green_go_AC4     EXIT              FIRST 1     dec num_green_go_AC_int4
 green_left_AC4 inters_int4       if num_abus_int4>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_AC_int4
                                     route 2
                                     end


                                                       1    dummy_green_left EXIT              FIRST 1
                                                       2    green_left_AC4   left_int4         FIRST 1
 green_left_AC4 left_int4         wait 16sec           1    green_left_AC4   EXIT              FIRST 1     dec
num_green_left_AC_int4
 green_go_BD4   inters_int4       if num_abus_int4>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_go_BD_int4
                                     route 2
                                     end               1    dummy_green_go   EXIT              FIRST 1
                                                       2    green_go_BD4     go_int4           FIRST 1
 green_go_BD4   go_int4           wait 16sec           1    green_go_BD4     EXIT              FIRST 1     dec num_green_go_BD_int4

 green_left_BD4 inters_int4       if num_abus_int4>0 then
                                  route 1
                                  else
                                     begin
                                     inc num_green_left_BD_int4
                                     route 2
                                     end               1    dummy_green_left EXIT              FIRST 1
                                                       2    green_left_BD4   left_int4         FIRST 1
 green_left_BD4 left_int4         wait 16sec           1    green_left_BD4   EXIT              FIRST 1     dec
num_green_left_BD_int4
 ABUS_A         inters_int4       time1 = clock()      1    ABUS_A           Abus_lane_AC_int4  FIRST 1
 ABUS_A         Abus_lane_AC_int4


                                                       1    ABUS_A           sensor_A_int4     FIRST 1
```

```
ABUS_A         sensor_A_int4     inc num_abus_int4
                                 inc num_abus_A_int4

                                                     1    ABUS_A           receive_C_int4,500 FIRST 1     time1 = clock() - time1
                                                                                                          var1int4 = time1
                                                                                                          move for 3sec

ABUS_A         receive_C_int4    inc num_abus_to_C_int4
                                 wait 5sec

                                 dec num_abus_to_C_int4
                                 dec num_abus_A_int4
                                 dec num_abus_int4
                                                     1    ABUS_A           EXIT               FIRST 1
ABUS_C         inters_int4       time1 = clock()     1    ABUS_C           abus_lane_CA_int4  FIRST 1
ABUS_C         abus_lane_CA_int4


                                                     1    ABUS_C           sensor_C_int4      FIRST 1


ABUS_C         sensor_C_int4     inc num_abus_int4
                                 inc num_abus_C_int4
                                                     1    ABUS_C           receive_A_int4     FIRST 1     move for 3sec
ABUS_C         receive_A_int4    inc num_abus_to_A_int4
                                 wait 5sec

                                 dec num_abus_to_A_int4
                                 dec num_abus_C_int4
                                 dec num_abus_int4
                                                     1    ABUS_C           inters_int3        FIRST 1     move for N(6,0.6)min
Car            start_A_int4                          1    Car              A_L_queue_int4,1   0.250000 1
                                                                           A_G_queue_int4,1   0.500000
                                                                           A_R_queue_int4,1   0.250000
Car            A_L_queue_int4                        1    Car              A_L_lane_int4      FIRST 1


Car            A_G_queue_int4                        1    Car              A_G_lane_int4      FIRST 1
Car            A_R_queue_int4                        1    Car              A_R_lane_int4      FIRST 1
Car            A_L_lane_int4     time_a=clock()

                                                     1    Car              receive_D_int4     FIRST 1     if
num_green_left_AC_int4<1 then
                                                                                                          wait until
num_green_left_AC_int4=1
```

```
                                                                                                   move for 2sec

                                                                                                   time_a=clock()-time_a
                                                                                                   varint4=time_a

 Car             A_G_lane_int4
                                                       1    Car              receive_C_int4     FIRST 1     if num_green_go_AC_int4<
1 then
                                                                                                   wait until
num_green_go_AC_int4=1
                                                                                                   else
                                                                                                     begin
                                                                                                       if
num_abus_to_B_int4>0 then
                                                                                                       wait until
num_abus_to_B_int4=0
                                                                                                     end

                                                                                                   move for 2sec
 Car             A_R_lane_int4                         1    Car              receive_B_int4     FIRST 1     if num_abus_to_B_int4>0
then

                                                                                                   wait until
num_abus_to_B_int4=0
                                                                                                   move for 2sec


 Car             start_B_int4                          1    Car              B_L_queue_int4     0.250000 1
                                                                             B_G_queue_int4     0.500000
                                                                             B_R_queue_int4     0.250000
 Car             B_L_queue_int4                        1    Car              B_L_lane_int4      FIRST 1
 Car             B_G_queue_int4                        1    Car              B_G_lane_int4      FIRST 1
 Car             B_R_queue_int4                        1    Car              B_R_lane_int4      FIRST 1
 Car             B_L_lane_int4                         1    Car              receive_A_int4     FIRST 1     if
num_green_left_BD_int4<1 then
                                                                                                   wait until
num_green_left_BD_int4=1

                                                                                                   move for 2sec

 Car             B_G_lane_int4                         1    Car              receive_D_int4     FIRST 1     if num_green_go_BD_int4<1
then
                                                                                                   wait until
num_green_go_BD_int4=1

                                                                                                   move for 2sec
```

```
 Car            B_R_lane_int4                       1    Car               receive_C_int4     FIRST 1     move for 2sec
 Car            start_C_int4                        1    Car               C_L_queue_int4     0.250000 1
                                                                           C_G_queue_int4     0.500000
                                                                           C_R_queue_int4     0.250000
 Car            C_L_queue_int4                      1    Car               C_L_lane_int4      FIRST 1
 Car            C_G_queue_int4                      1    Car               C_G_lane_int4      FIRST 1
 Car            C_R_queue_int4                      1    Car               C_R_lane_int4      FIRST 1
 Car            C_L_lane_int4                       1    Car               receive_B_int4     FIRST 1     if
num_green_left_AC_int4<1 then
                                                                                                          wait until
num_green_left_AC_int4=1

                                                                                                          move for 2sec


 Car            C_G_lane_int4                       1    Car               receive_A_int4     FIRST 1     if num_green_go_AC_int4<
1 then
                                                                                                          wait until
num_green_go_AC_int4=1

                                                                                                          else
                                                                                                            begin
                                                                                                              if
num_abus_to_D_int4>0 then
                                                                                                              wait until
num_abus_to_D_int4=0
                                                                                                            end

                                                                                                          move for 2sec

 Car            C_R_lane_int4                       1    Car               receive_D_int4     FIRST 1     if num_abus_to_D_int4>0
then

                                                                                                          wait until
num_abus_to_D_int4=0

                                                                                                          move for 2sec


 Car            start_D_int4                        1    Car               D_L_queue_int4     0.250000 1
                                                                           D_G_queue_int4     0.500000
                                                                           D_R_queue_int4     0.250000
 Car            D_L_queue_int4                      1    Car               D_L_lane_int4      FIRST 1
 Car            D_G_queue_int4                      1    Car               D_G_lane_int4      FIRST 1
 Car            D_R_queue_int4                      1    Car               D_R_lane_int4      FIRST 1
```

```
 Car            D_L_lane_int4                         1    Car              receive_C_int4     FIRST 1     if
num_green_left_BD_int4<1 then
                                                                                                           wait until
num_green_left_BD_int4=1

                                                                                                           move for 2sec


 Car            D_G_lane_int4                         1    Car              receive_B_int4     FIRST 1     if num_green_go_BD_int4<1
then
                                                                                                            wait until
num_green_go_BD_int4=1

                                                                                                           move for 2sec

 Car            D_R_lane_int4                         1    Car              receive_A_int4     FIRST 1     move for 2sec
 Car            receive_A_int4                        1    Car              start_C_int3       FIRST 1     move for N(6,0.6)min
 Car            receive_B_int4    if car_ID=4 then
                                  begin
                                   inc num_cross_int4
                                   road_time_4=clock()-road_time_4
                                   total_road_time_int4=total_road_time_int4+road_time_4
                                   avg_road_time_int4=total_road_time_int4/num_cross_int4
                                  end                 1    Car              EXIT               FIRST 1
 Car            receive_C_int4    if car_ID=0 then
                                  begin
                                   road_time_0=clock()-road_time_0
                                   total_road_time_0=total_road_time_0+road_time_0
                                   inc num_through_0
                                   avg_road_time_0=total_road_time_0/num_through_0
                                  end
                                                      1    Car              EXIT               FIRST 1
 Car            receive_D_int4    if car_ID=4 then
                                  begin
                                   inc num_cross_int4
                                   road_time_4=clock()-road_time_4
                                   total_road_time_int4=total_road_time_int4+road_time_4
                                   avg_road_time_int4=total_road_time_int4/num_cross_int4
                                  end                 1    Car              EXIT               FIRST 1



*******************************************************************************
*                                   Arrivals                                  *
```

```
****************************************************************************************

  Entity         Location     Qty each   First Time Occurrences Frequency    Logic
  -------------- ------------ ---------- ---------- ----------- ------------ ------------
  Car            start_A_int1 1          0          inf         E(2)sec      car_ID=0
                                                                             road_time_0=clock()
  Car            start_B_int1 1          0          INF         E(2)sec      car_ID=1
                                                                             road_time_1=clock()
  Car            start_D_int1 1          0          INF         E(2)sec      car_ID=1
                                                                             road_time_1=clock()
  Car            start_B_int2 1          0          INF         E(8)sec      car_ID=2
                                                                             road_time_2=clock()
  Car            start_D_int2 1          0          INF         E(8)sec      car_ID=2
                                                                             road_time_2=clock()
  Car            start_B_int3 1          0          INF         E(8)sec      car_ID=3
                                                                             road_time_3=clock()
  Car            start_D_int3 1          0          INF         E(8)sec      car_ID=3
                                                                             road_time_3=clock()
  Car            start_B_int4 1          0          INF         E(2)sec      car_ID=4
                                                                             road_time_4=clock()
  Car            start_D_int4 1          0          INF         E(2)sec      car_ID=4
                                                                             road_time_4=clock()
  Car            start_C_int4 1          0          INF         E(2)sec      car_ID=0
                                                                             road_time_0=clock()
  ABUS_A         inters_int1  1          0          INF         N(15,0.5)min

  ABUS_C         inters_int4  1          0          INF         N(15,0.5)min
  green_left_AC1 inters_int1  1          54sec      INF         72sec


  green_go_AC1   inters_int1  1          0sec       INF         72sec




  green_left_BD1 inters_int1  1          18sec      INF         72sec
  green_go_BD1   inters_int1  1          36sec      INF         72sec
  green_left_AC2 inters_int2  1          42sec      INF         60sec
  green_go_AC2   inters_int2  1          0sec       INF         60sec




  green_left_BD2 inters_int2  1          18sec      INF         60sec
  green_go_BD2   inters_int2  1          30sec      INF         60sec
  green_left_AC3 inters_int3  1          42sec      INF         60sec
```

```
  green_go_AC3   inters_int3  1          0sec       INF         60sec



  green_left_BD3 inters_int3  1          18sec      INF         60sec
  green_go_BD3   inters_int3  1          30sec      INF         60sec
  green_left_AC4 inters_int4  1          54sec      INF         72sec


  green_go_AC4   inters_int4  1          0sec       INF         72sec




  green_left_BD4 inters_int4  1          18sec      INF         72sec
  green_go_BD4   inters_int4  1          36sec      INF         72sec

********************************************************************************
*                                  Attributes                                  *
********************************************************************************

  ID          Type         Classification
  ----------- ------------ --------------
  time1       Real         Entity
  time2       Real         Entity
  Time3       Real         Entity
  time_a      Real         Entity
  car_ID      Integer      Entity
  road_time_0 Real         Entity
  road_time_1 Real         Entity
  road_time_2 Real         Entity
  road_time_3 Real         Entity
  road_time_4 Real         Entity

********************************************************************************
*                              Variables (global)                              *
********************************************************************************

  ID                     Type         Initial value Stats
  ---------------------- ------------ ------------- -----------
  num_abus_int1          Integer      0             Time Series
  num_abus_A_int1        Integer      0             Time Series
  num_abus_C_int1        Integer      0             Time Series
```

num_abus_to_D_int1	Integer	0	Time Series
num_abus_to_C_int1	Integer	0	Time Series
num_abus_to_B_int1	Integer	0	Time Series
num_abus_to_A_int1	Integer	0	Time Series
num_green_go_AC_int1	Integer	0	Time Series
num_green_go_BD_int1	Integer	0	Time Series
num_green_left_AC_int1	Integer	0	Time Series
num_green_left_BD_int1	Integer	0	Time Series
var1	Real	0	Time Series
var2	Real	0	Time Series
var3	Real	0	Time Series
var	Real	0	Time Series
num_abus_int2	Integer	0	Time Series
num_abus_A_int2	Integer	0	Time Series
num_abus_C_int2	Integer	0	Time Series
num_abus_to_D_int2	Integer	0	Time Series
num_abus_to_C_int2	Integer	0	Time Series
num_abus_to_B_int2	Integer	0	Time Series
num_abus_to_A_int2	Integer	0	Time Series
num_green_go_AC_int2	Integer	0	Time Series
num_green_go_BD_int2	Integer	0	Time Series
num_green_left_AC_int2	Integer	0	Time Series
num_green_left_BD_int2	Integer	0	Time Series
var1int2	Real	0	Time Series
var2int2	Real	0	Time Series
var3int2	Real	0	Time Series
varint2	Real	0	Time Series
num_abus_int3	Integer	0	Time Series
num_abus_A_int3	Integer	0	Time Series
num_abus_C_int3	Integer	0	Time Series
num_abus_to_D_int3	Integer	0	Time Series
num_abus_to_C_int3	Integer	0	Time Series
num_abus_to_B_int3	Integer	0	Time Series
num_abus_to_A_int3	Integer	0	Time Series
num_green_go_AC_int3	Integer	0	Time Series
num_green_go_BD_int3	Integer	0	Time Series
num_green_left_AC_int3	Integer	0	Time Series
num_green_left_BD_int3	Integer	0	Time Series
var1int3	Real	0	Time Series
var2int3	Real	0	Time Series
var3int3	Real	0	Time Series
varint3	Real	0	Time Series
num_abus_int4	Integer	0	Time Series
num_abus_A_int4	Integer	0	Time Series
num_abus_C_int4	Integer	0	Time Series
num_abus_to_D_int4	Integer	0	Time Series
num_abus_to_C_int4	Integer	0	Time Series

num_abus_to_B_int4	Integer	0	Time Series
num_abus_to_A_int4	Integer	0	Time Series
num_green_go_AC_int4	Integer	0	Time Series
num_green_go_BD_int4	Integer	0	Time Series
num_green_left_AC_int4	Integer	0	Time Series
num_green_left_BD_int4	Integer	0	Time Series
var1int4	Real	0	Time Series
var2int4	Real	0	Time Series
var3int4	Real	0	Time Series
varint4	Real	0	Time Series
total_road_time_0	Real	0	Time Series
avg_road_time_0	Real	0	Time Series
num_through_0	Integer	0	Time Series
total_road_time_int1	Real	0	Time Series
avg_road_time_int1	Real	0	Time Series
num_cross_int1	Integer	0	Time Series
total_road_time_int2	Real	0	Time Series
avg_road_time_int2	Real	0	Time Series
num_cross_int2	Integer	0	Time Series
total_road_time_int3	Real	0	Time Series
avg_road_time_int3	Real	0	Time Series
num_cross_int3	Integer	0	Time Series
total_road_time_int4	Real	0	Time Series
avg_road_time_int4	Real	0	Time Series
num_cross_int4	Integer	0	Time Series

EVALUATION OF BUS AND TRUCK AUTOMATION OPERATIONS CONCEPTS

PART IV: EVALUATION OF INTER-CITY TRUCK AUTOMATION

AN AUTOMATED HIGHWAY SYSTEM FOR INTER-CITY TRUCKING WITH SHUTTLE-CENTERED CONVOYING: OPERATIONS AND EVALUATION

H.-S. Jacob Tsao and Lan Zhang
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA

TABLE OF CONTENTS

AN AUTOMATED HIGHWAY SYSTEM FOR INTER-CITY TRUCKING WITH SHUTTLE-CENTERED CONVOYING: OPERATIONS AND EVALUATION

H.-S. Jacob Tsao and Lan Zhang
Department of Industrial and Systems Engineering
San Jose State University
San Jose, California 94583-0085, USA

ABSTRACT

Various intelligent transportation systems (ITS) operating concepts have been proposed for improving efficiency of truck operations. The most technology-intensive among them is an automated highway systems (AHS) dedicated to fully automated truck operations. The concept of automated highway systems (AHS) has been primarily motivated by the rapidly worsening traffic congestion on metropolitan highways and the potential of AHS for drastically increasing vehicle throughput in the existing right-of-way. The overwhelming majority of the research has been focused on automobile-AHS. This paper focuses instead on the automation of inter-city trucking for the purpose of increasing trucking productivity, of which vehicle throughput is only one of many factors. Based on customer needs, stakeholder concerns and available or promising truck-automation technologies, we developed for the Phase I of this research design options for several key aspects of truck-AHS operations, compared the merits of these options, and developed system operating concepts and deployment sequences to satisfy the customer needs. Based on an initial qualitative analysis, the phase-I research developed two operating concepts. This research developed a new operating concept. In this new operating concept, the system is closed in the sense that tractors traveling on the truck-AHS are provided by a small number of operators called "AHS haulers." Also, trucks form a closely-spaced convoy while traveling automatically on the AHS; only the lead truck of the convoy has a human driver, who supervises the operations of the whole convoy. Moreover, convoy merging at an on-ramp and convoy splitting at an off-ramp are both automated. This research further developed this new operating concept to include sufficient operational details for quantitative evaluation and comparison, evaluated its merits and compared it to two conventional alternatives. In quantitative evaluation and comparison, we focus on truck travel time, non-truck travel time, trucking labor and fuel consumption.

Based on our results, it is clear that, under the assumptions made and given the corridor selected, the truck-AHS alternative will not provide any travel time advantage to the overall system or even to the overall trucking industry. The only advantages of truck-AHS are labor and fuel savings, and the labor saving is drastic. These savings must be weighed against the infrastructure costs. Our comparison suggests that the general-use-lane alternative is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings. Given our results and their limitations, future studies on truck-AHS should be focused on the feasibility of driverless, automated, closely-spaced truck-following as potential source of labor and fuel savings and on the concomitant infrastructure costs if the purpose of constructing a truck-AHS is to facilitate inter-city trucking. Other truck-AHS operating concepts may benefit other more special purposes.

AN AUTOMATED HIGHWAY SYSTEM FOR INTER-CITY TRUCKING WITH SHUTTLE-CENTERED CONVOYING: OPERATIONS AND EVALUATION

EXECUTIVE SUMMARY

Various intelligent transportation systems (ITS) operating concepts have been proposed for improving efficiency of truck operations. The most technology-intensive among them is an automated highway systems (AHS) dedicated to fully automated truck operations. The concept of automated highway systems (AHS) has been primarily motivated by the rapidly worsening traffic congestion on metropolitan highways and the potential of AHS for drastically increasing vehicle throughput in the existing right-of-way. The overwhelming majority of the research has been focused on automobile-AHS. This paper focuses instead on the automation of inter-city trucking for the purpose of increasing trucking productivity, of which vehicle throughput is only one of many factors. Based on customer needs, stakeholder concerns and available or promising truck-automation technologies, we developed for the Phase I of this research design options for several key aspects of truck-AHS operations, compared the merits of these options, and developed system operating concepts and deployment sequences to satisfy the customer needs. Based on an initial qualitative analysis, the phase-I research developed two operating concepts. Those concepts resemble commercial rail operations in the train-like operations. But they differ from the rail operations considerably in that they are implemented on a long stretch of dedicated and physically separated freeway, possibly occupying the median space of the current interstate freeway system, along a busy freight corridor and that (self-propelled) dual-mode trucks are electronically coupled and organized into convoys while traveling on the freeway, instead of mechanically coupled, and hence can move onto or off from the freeway with ease. We refer to this scenario as Truck Automated Highway System (Truck-AHS). This research developed a new operating concept called Truck-AHS with shuttle-centered convoying; the new concept was motivated to respond to user needs and to increase the deployability while taking advantage of promising technologies.

In this new operating concept, the system is closed in the sense that tractors traveling on the truck-AHS are provided by a small number of operators called “AHS haulers.” Also, trucks form a closely-spaced convoy while traveling automatically on the AHS; only the lead truck of the convoy has a human driver, who supervises the operations of the whole convoy. Moreover, convoy merging at an on-ramp and convoy splitting at an off-ramp are both automated. This research further developed this new operating concept to include sufficient operational details for quantitative evaluation and comparison. For example, the Truck-AHS operator runs a shuttle truck from one end of the corridor to the other and back; such a shuttle truck serves as the lead truck of a truck convoy and only such a truck can be the lead truck of a truck convoy. Any automated truck along the corridor that wishes to use the Truck-AHS must join a truck convoy already traveling on the Truck-AHS. The headway of the shuttles is constant throughout the day; the constant headway is determined in such a way that any truck arriving at an Truck-AHS access point can join the next passing truck convoy (after changing modes and waiting for the next passing convoy) and the convoy size does not exceed a prescribed number of trucks, e.g., the limit of 25 trucks in our numerical study. The trucks of a convoy are closely spaced and hence enjoy fuel savings due to reduced air resistance.

We compared the following three specific alternatives in terms of truck travel time, non-truck travel time, trucking labor and fuel consumption:

- General-Use Lane (adding one conventional general-use lane per direction to the conventional freeway)
- Truck-AHS (constructing a physically separated one-lane truck-AHS within or along the right-of-way of a conventional freeway)
- Truck Lane (constructing a physically separated lane dedicated to truck travel within or along the right-of-way of a conventional freeway).

Based on our numerical results, it is clear that, under the assumptions made and given the corridor selected, the Truck-AHS alternative will not provide any travel time advantage to the overall system or even to the overall trucking industry. It is able to provide travel time advantage to only the long-haul trucking industry but at the expense of the short-haul trucking industry and the non-truck driving public. (Trucks using the truck-AHS travel at a consistently higher speed of 75 miles per hour.)

The only advantages of truck-AHS are labor and fuel savings, and the labor saving is drastic. These savings must be weighed against the infrastructure costs. Moreover, the safety and technical feasibility of the truck-AHS must be carefully studied. The Truck Lane alternative does not look promising either; it does not enjoy any advantage over the General-Use Lane alternative, and we have not even begun to address the cost of infrastructure.

Finally, for the General-Use Lane alternative, we compared the addition of only one conventional general-use lane to the other two alternatives. After the "overhead" infrastructure requirements, e.g., the break-down lane or shoulder and the width required for physical barriers, etc., for the other two alternatives are taken into consideration, it is likely that the overall right-of-way required by either of the other two alternatives can accommodate two conventional general-use lanes. As a result, the travel time advantages of the General-Use Lane alternative just reported will likely be clear understatements. In addition, addition of two general-use lanes, without the need for physical separation as required by the other two alternatives, will likely cost drastically less.

Our comparison suggests that general-use lane is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings.

Although these results can provide valuable insights into the advantages and disadvantages of the three alternatives in general, the reader is reminded that what we have evaluated and compared are three specific operational systems and that the comparison is made against a specific reference corridor. In addition, due to the complexity of the problem and the absence of daily origin-destination data for truck trips and time-dependent demand data for freeway sections, several estimation methods have been employed. Caution is needed when generalizing these specific results to the three corresponding general alternatives.

Based on our results and given these limitations, we believe that future studies on truck-AHS should be focused on the feasibility of driverless, automated, closely-spaced truck-following as potential source of labor and fuel savings and on the concomitant infrastructure costs if the purpose of constructing a truck-AHS is to facilitate inter-city trucking. Other truck-AHS operating concepts may benefit other more special purposes.

AN AUTOMATED HIGHWAY SYSTEM FOR INTER-CITY TRUCKING WITH SHUTTLE-CENTERED CONVOYING: OPERATIONS AND EVALUATION

1. INTRODUCTION

The concept of automated highway systems (AHS) has been primarily motivated by the rapidly worsening traffic congestion on metropolitan highways and the potential of AHS to drastically increase vehicle throughput, particularly automobile throughput, without requiring a significant amount of additional right-of-way. AHS-related research has been focused on automobile-AHS. Various operating concepts have been developed for an "end-state AHS" (e.g., Sengupta et al., 1996). Al-Ayat and Hall (1994), Tsao (1995b, 1995c, 1995d, 1998b), Hall and Tsao (1997a), AlKadri, Benouar, and Tsao (1999), Shladover (2000) and others also studied the deployment issues and proposed deployment sequences for such end-state AHS.

This paper focuses on the automation of inter-city trucking for the purpose of increasing trucking productivity, of which vehicle throughput on highways is only one of many factors. Several systems studies investigating truck automation were funded under the AHS Precursor Systems Analysis (PSA) Program (Calspan, 1995). DaimlerChrysler has been developing and testing the technologies enabling automated truck convoying (Riva and Ulken, 1997; Borodani et al., 1997; Schulze, 1997; Ulmer, 1999) as well as conducting economic evaluation of the technologies (Baum and Schulz, 1997). Recently, California Partners for Advanced Transit and Highways (PATH) has also investigated into advanced control technology for heavy vehicles, e.g., (Tai, 2001; Tai et al. 2001)

The previous phase, i.e., Phase I, of this research was motivated to (a) identify opportunities for automation technologies to benefit *heavy-vehicle operations*, (b) develop operating concepts in sufficient detail for specifying *vehicle and infrastructure functional requirements*, and (c) develop *deployment sequences* for these operating concepts. As part of the Phase-I research, Tsao and Botha (2002b, 2003) identified the needs of the long-haul trucking industry and the major concerns of key stakeholders. Based on those customer needs and stakeholder concerns and available or promising truck-automation technologies, they developed design options for several key aspects of truck-AHS operations. After comparing the relative merits of these options, they also developed two operating concepts and the corresponding deployment sequences to satisfy the customer needs.

The purposes of this research are to (a) further develop truck-AHS operating concepts, (b) develop sufficient *operational details* in order to perform cost-benefit evaluations and comparisons with conventional alternatives and (c) to perform the *evaluations and comparisons*.

In Phase I, Tsao and Botha (2002b) developed two operating concepts. This phase-II research developed a new operating concept. In this new operating concept, the system is closed in the sense that tractors traveling on the truck-AHS are provided by a small number of operators called "AHS haulers." Also, trucks form a closely-spaced convoy while traveling automatically on the AHS; only the lead truck of the convoy has a human driver, who supervises the operations of the whole convoy. Moreover, merging at an on-ramp and splitting at an off-ramp are both automated. This research further developed this new closed-system concept to include sufficient operational details for quantitative evaluation and comparison, evaluated its merits and compared it to two conventional alternatives. The two conventional alternatives are addition of a general-use lane and addition of a truck lane. In quantitative evaluation and comparison, we focus on truck travel time, non-truck travel time, truck labor and fuel consumption. To make the evaluation and comparison more realistic, we selected the California portion of the Interstate 5 for reality checking and developed a reference corridor based on it.

The rest of this paper is organized as follows. Section 2 describes the new operating concept and discusses the operational details for quantitative evaluation and comparison. Section 3 describes the scope of comparison, a reference freight corridor, performance measures and the actual comparison scenarios. Section 4 addresses the evaluation and comparison methodology. Section 5 briefly describes the software tools we have developed. Section 6 summarizes the comparison results. Concluding remarks are given in Section 7. Appendix A provides details about the reference corridor. Appendix B contains the detailed requirements document for the computer tools. Appendix C contains the program listings of the software tools developed according to the requirements specified in Appendix B.

2. A CLOSED-SYSTEM OPERATING CONCEPT WITH DYNAMIC CLOSELY-SPACED CONVOYING FOR A PROTECTED INTER-CITY TRUCK-AHS

In this section, we first describe the key design features of a new truck-AHS operating concept. After discussing the key advantages and disadvantages as the motivation of this concept, we describe the system operations and some detailed operating rules.

2.1 Vehicle-System Design Features

This system operating concept features the following design options: (1) closely-spaced truck convoying enabled by electronic coupling, (2) automated driving supervised by a human driver in the lead truck and driverless truck-following, (3) closed system and (4) dynamically infrastructure-supervised, i.e, driverless, merging and splitting only at or near on- and off-ramps, respectively. The first two options are self-explanatory. We briefly describe the last two, which combined are actually the key features that distinguish this new concept from the two developed in Phase I.

- *Closed System*: The closeness or openness of a truck-AHS has to do with the institutional structure of AHS operations, particularly regarding provision of the line-haul service and the local collection-and-distribution service. In a closed system, the line-haul service and the local collection-and-distribution service are allowed to be performed by separate companies: AHS haulers and local feeder haulers. Tsao and Botha (2001, 2002a, 2002b) refer to a carrier offering the service of hauling other companies' trucks or just trailers on a truck-AHS, in addition to its own freight, as an AHS Hauler. Note that the truck-AHS is open only to the tractors operated by the AHS haulers. The exclusive use by the AHS haulers disqualifies the system as an *open system*.
- *Dynamic Convoying - Infrastructure-Supervised Merging and Splitting Only at or Near On- and Off-Ramps, Respectively*: A convoy entering the mainline may join another convoy already traveling on the mainline by merging with the mainline convoy from behind. However, convoy merging may occur only for this purpose and hence may occur only at or near a location where an on-ramp and the mainline merge. A portion of a

convoy already traveling on the mainline may split off from the convoy to exit the mainline. However, convoy splitting may occur only for this purpose and hence may occur only at or near a location where the mainline splits to accommodate an off-ramp. A driver is not required for the lead truck of either the entering convoy or the exiting portion of a mainline convoy.

2.2 Motivation for the New Operating Concept

To motivate this new operating concept, we focus on the two key features described in Section 2.1. The feature of closed-system is motivated by the deployment issues of a truck-AHS. The feature of dynamic convoying is motivated by operational efficiency and even safety. We focus on only advantages or disadvantages not identified or discussed in Tsao and Botha (2002b, 2003).

Some critical issues in deploying AHS and some specific criteria for evaluating AHS deployment strategies can be found in (Al-Ayat and Hall, 1994; Tsao, 1995b and 1995c; Hall and Tsao, 1997; Hall, 1997; AlKadri et al., 1998; Tsao, 1998b; Tsao, 2001). Critical truck-AHS deployment issues include the requirement of a physically separated and dedicated truck lane for safety, the requirement of a sufficiently large population of vehicles equipped with advanced technology to avoid the "empty-lane syndrome," the "chicken-and-egg" issue" resulting from these two requirements, the long-haul industry's requirement for a fast return on investment (ROI), the risk of opposition from the current user of the public right-of-way (e.g., opposition to disallowing use of a truck-lane by conventional trucks after converting the truck-lane to AHS, the opposition to taking away an existing general-use lane for exclusive truck use, the opposition to taking away the right-of-way for possible future allocation for general use), competition or opposition from other modes of freight transportation (e.g., intermodal rail, etc.), human-factors issues, liability issues, etc. For a general framework for evaluating ITS deployment strategies, the reader is referred to (Tsao, 2001).

A major difficulty in AHS research is the so-called "the chicken-and-egg problem," i.e., building infrastructure first to entice the purchase by the trucking industry of a sufficient quantity of equipped vehicles or building vehicle population first so as to justify the infrastructure construction and avoid the "the empty-lane syndrome" once the infrastructure is built (Tsao 1995a, 1995c, 1998a, 2001). Several efforts in tackling this issue have been reported in the literature. Specific deployment sequences for a general-use AHS have been proposed (Tsao, 1995a); such sequences have also been proposed for a truck-AHS (Tsao and Botha, 2003). Almost all of these efforts focused on an open system that is to be "filled" with equipped vehicles operated by the general freight carriers. This paper proposes a new operating concept and calls for building the infrastructure and vehicle population simultaneously in an attempt to avoid the chicken-and-egg problem. This is similar to the railroad operations in the sense that a railroad is a closed system, is operated by the owner of a railroad, and is possibly used by another railroad operator on a fee-for-use basis. However, this is different from the railroad operations in the sense that the right-of-way is public, although the concept can be readily transformed into one involving private right-of-way.

A closed truck-AHS has the potential of avoiding many issues accompanying an open truck-AHS. First of all, as mentioned in Tsao and Botha (2003), a close truck-AHS does not require a large population of trucks that are equipped with the required technology, and may allow much faster deployment of a truck-AHS and reap much earlier the benefit of automation than an open system. The safety and the operational reliability of the system can be enhanced by the small number of fleets, and standards are more easily established and adhered to among the small number of fleet operators. Unlike an open system, the driver performing the drayage when arriving at the access point of the truck-AHS knows for certain that there will be a AHS-hauler truck that will haul the trailer to the destination on the truck-AHS and that the driver will not be needed for mainline travel. Other advantages include liability resolution, etc. Another advantage of this system is that it may be used to simply to haul containers off-loaded from a container ship at a seaport through a highly congested metropolitan area to an inland location for a possible mode change for further movement. A major disadvantage is the need for a mode change to transfer a trailer between a conventional tractor and an AHS tractor, which requires special equipment and additional operations. There may also be liability complications because of the change of haulers. However, the railroad industry has dealt with such complications routinely for decades.

A major issue associated with an open truck AHS is speed uniformity, at least among the trucks within a convoy. It is well known that top truck speed varies with respect to hauling power, which varies widely among trucks. Also, top truck speed varies with respect to weight of freight greatly for any give truck. The speed of a convoy will have to be the top speed of the slowest truck of the convoy. In addition, in an open system where there is only one lane, the speed of all the convoys traveling on the lane can only be that of the slowest convoy. This could be a serious issue, and to overcome this issue, a second truck lane or at least a periodic passing lane may be required. However, such additional lane will still not overcome the issue of convoy speed being limited by the slowest truck of the convoy. With a closed system, only one AHS lane is required, and no passing lane is required because the system is operated by a small number of AHS haulers. For the same reason, a uniform top speed can be much more easily achieved and sustained than an open system.

A major advantage of Dynamic Convoying - Infrastructure-Supervised Merging and Splitting Only at On- and Off-Ramps, Respectively is to avoid the disturbance that would otherwise occur if every exiting truck has to first split from the rest of the truck of the convoy on the mainline in the sense that it achieves a distance comparable to the safe distance between two trucks operating on the conventional highway. Note that such splitting if necessary would cause significantly more disturbance to the convoy operations if more trucks in the convoy needs to exit. It is also to avoid the disturbance that would otherwise occur if every entering convoy has to enter the mainline as a separate convoy and possibly then merge with a convoy already traveling on the mainline in front or in rear of it.

2.3 System Operations

We summarize normal operations and operations related to abnormal events of this closed system. We focus on AHS operations only; their integration with local feeder operations is omitted.

Normal Operations:

Normal Mainline Operations

- A single lane in each direction physically separated from manual traffic, without a full breakdown lane but with a shoulder that is sufficiently wide so that the single lane plus the shoulder are able to accommodate one disabled truck and one lane of through traffic at a moderate speed. Note that the shoulder will likely be needed for other purposes also. For example, tire treads separated from truck tires may move or be moved onto the shoulder. In such a case, the traffic will not be impeded by the presence of such large debris. Where additional right-of-way is available, a full second lane can be provided as or a breakdown lane. (As mentioned earlier, no passing lane is required because the system is operated by a small number of AHS haulers, and uniform top speed of all trucks can be much more easily achieved and sustained.)
- Automated vehicle control, including lateral and longitudinal control, enabling hands-off and feet-off truck operation
- Automated convoying, with an upper limit on the length of a convoy: The lead truck of a convoy must have a driver. The driver of the lead truck is responsible for detecting debris ahead on the lane or other abnormal events that cannot be reliably or cost-effectively detected by automation; the driver may also be tasked with actually driving the truck, with or without the assistance of automation.
- Driverless trailing trucks: No drivers are required on the trailing trucks of an automated convoy.
- Closely-spaced convoying: The shorter the distance between two trucks in a convoy, the less "wind-drag" on both trucks and hence the higher fuel efficiency. Moreover, the shorter the distance, the higher the mainline capacity. The achievable minimum safe distance is a subject of future research. Also, the shorter the distance, the lower impact speed if two longitudinally adjacent trucks collide in an accident (Tsao and Hall, 1994).

Normal Access and Egress Operations:

- Dedicated on- and off-ramps and staging areas: Design of staging areas will require future research.
- Multi-destination convoying: A convoy may consist of trucks destined for different exits.
- Convoy entry into the AHS mainline after assembled in an staging area
- Automated convoy merging at on-ramp: An entering convoy or truck may "tag" onto the end of a convoy already traveling on the mainline at or near an on-ramp, without first entering the AHS mainline as a separate convoy (with long inter-convoy distances from the longitudinally adjacent convoys) and then merge with a neighboring convoy. This feature may reduce disturbance to mainline traffic at access locations and increase the mainline capacity (Hall and Tsao, 1997b; Hall, Nowroozi and Tsao, 2001; Tsao, Hall and Chatterjee, 1997). Such tagging is a form of convoy merging, but is performed at or near an on-ramp. With on-ramp areas closely monitored by the infrastructure for possible safety-impacting debris or events, such convoy-tagging requires no driver on the lead truck of the entering convoy, and hence reduces labor requirements.
- Convoy exiting into a staging area at the destination exit.
- Automated convoy-splitting at off-ramp: Convoy splitting to facilitate exiting of a portion of the convoy: Without such splitting, the whole convoy would have to exit the mainline to let

the exiting trucks leave the convoy and hence may cause congestion at staging areas under some conditions and delay to the trucks that are destined for exits ahead. Such convoy splitting is performed only at or near an off-ramp. The lead truck of the exiting portion needs not have a driver; the splitting operation is supervised by the infrastructure, which monitors the off-ramp area closely.

- "Simultaneous splitting and exiting": A portion of a convoy splits away from the convoy as it exits the mainline and moves into the off-ramp. In other words, the exiting portion need not separate itself from the rest of the convoy at full inter-convoy distances before moving into the off-ramp. This feature may also reduce disturbance to mainline traffic at egress locations and increase the mainline capacity (Hall and Tsao, 1997b; Hall, Nowroozi and Tsao, 2001).

Operations Related to Abnormal Events (Issues and Solutions)

- The driver of the lead truck of a convoy watches for possible safety hazards, e.g., obstacles or large debris, for the whole convoy. (Replacing human cognitive ability and adaptability by machine is difficult.)
- To the extent possible, organizing trucks of different characteristics into different convoys to minimize probability and severity of intra-convoy collision, e.g., braking capability, etc. Organizing trucks of different destinations (at the origin staging) into different convoys to maximize operational efficiency
- A disabled truck should be parked on the shoulder, if possible. Traffic will have to slow down and use the remaining space to bypass the disabled truck; all these and other necessary maneuvers are automated.
- Intra-convoy collision would involve lesser liability issues because all trucks in a convoy are operated by the same or a small number of companies.

2.4 Operating Rules: Routing and Scheduling

We now describe a small number of operating rules. These rules are not optimized, but are included to enable quantitative evaluation and comparison.

Rule 1: End-to-end Shuttle Convoy; Fixed Headway: Run an end-to-end shuttle truck with a driver aboard between the two end points with a fixed headway. In our case, run a shuttle between the California-Mexico border and the California-Oregon border. The headway is selected so that the maximum convoy size, i.e., the maximum number of trucks in a convoy, is not excessive. In our study, 20 is considered a target maximum size, but in some busy AHS sections, the convoy size is actually larger than 20. Although the end-to-end shuttle trucks may haul freight, the amount of freight to be hauled by these shuttle trucks depends on the freight demand. In this study, the freight demand is not sufficient to occupy all such shuttle trucks, and, we therefore, assume that none of such end-to-end shuttle trucks actually haul any freight. As a consequence, such shuttle trucks can be viewed as "overhead," and their use incurs additional travel time that the other two alternatives do not incur at all. (It is conceivable that given the capacity and given the "overhead" nature of the shuttle runs, the AHS operator could easily attract sufficient load to occupy all the capacity by lowering the charges.)

Rule 2: All trucks traveling in a Convoy, with a Driver onboard only the Lead Truck: All trucks traveling on the truck-AHS must be attached to a closely spaced convoy, and every such convoy must be led by an end-to-end shuttle truck. Truck-following is automated and is driverless.

Rule 3: Maximum Convoy Size, with Flexibility for Larger Size to Cope with Demand Volatility. For safety reasons, set a maximum convoy size. As just mentioned, we use 20 as the target maximum size. However, to cope with possibly volatile demand, actual size of a convoy may be allowed to exceed 20 depending on circumstances.

Rule 4: An Entering Truck or Convoy Joins the Next Passing Shuttle Convoy. Trucks arriving at a truck-AHS access point will join the next passing shuttle convoy on the mainline after a proper mode change. If there are more than one truck that wish to enter the AHS, organize these trucks into a convoy and the convoy will enter the AHS by joining the next passing truck convoy already traveling on the AHS.

3. EVALUATION AND COMPARISON

This section describes the scope of comparison, a reference freight corridor, performance measures and the actual comparisons.

3.1 Scope of Comparison

We evaluate and compare the benefit and cost of the following three alternatives:

- Adding an exclusive AHS truck lane
- Adding a dedicated conventional truck lane
- Adding a conventional general-use lane (without dedication of any lanes to truck use).

Adding a dedicated conventional truck lane involves several design options and many complicated issues. A design option is about physical separation, which has implications on safety, operational efficiency for trucks, disturbance to the other traffic, cost, etc. For a physically separate system, two lanes are required because of the wide range of possible truck speeds. At least, long and frequent passing lanes must be equipped. Such a system requires much right-of-way, and we consider such a system as possible only for short stretches of the current freeway system of the nation, at least in California.

Without physical separation, there are several options. The options include dedication of two truck lanes on the left (i.e., dedication of two truck lanes next to the median) and dedication of one truck lane next to the right-most lane, with transient use by non-trucks for entry into general-use lanes or for exiting to off-ramps . (The right-most lane is used by all traffic for entry and exiting, and hence cannot be used as dedicated or virtually dedicated lane.) The dedication of two lanes in the former is motivated by the issue of truck top speed and the resulting necessity for a second or a passing lane. Without the two lanes, passing is either not allowed or may be done by using the adjacent general-use lane, which will be the fast lane for the rest of the traffic. This creates safety hazards. However, we believe that the right-of-way requirement for such two truck lanes along a long stretch of freeway makes it virtually impossible to implement, at least in California.

However, the latter has its own problems. First of all, user fees will be difficult to assess because trucks traveling on the right-most lane still need to use to the truck lane (i.e., the lane next to the right-most lane) for yielding to entering traffic for safety and because trucks not intending to use the truck lane still need to pass slow trucks ahead of them. In addition, with the difficult-to-enforce or even un-enforceable restriction for the truck lane to only truck usage, the benefit for a truck to use such a truck lane is unclear.

Therefore, given these critical issues, we consider the following truck lane option. The option involves an infrastructure that is identical to that of a truck-AHS but allows a slower speed. The slower speed is motivated by at least two considerations. First, the operational uniform speed of a truck AHS may be too difficult to achieve and sustain for all or most trucks that could benefit from the truck lane. Second, the conventional trucks would not be equipped with the automated lane-keep function that all trucks using a truck-AHS would have, and hence could not operate safely at the truck-AHS speed.

Performance aspects compared include truck travel time, non-truck travel time, trucking labor, trucking fuel consumption and equipment requirement. Since truck travel time is a good surrogate for truck equipment requirement, we explicit track the former, but not the latter.

3.2 A Reference Freight Corridor

To ensure development of realistic operating concepts, their evaluation and their comparisons with other alternatives, we developed a reference freight corridor based on the California portion of Interstate 5, i.e., the portion of Interstate 5 from California-Mexico border to the Oregon-California border.

In essence, the evaluation and comparison is decoupled into two separate and parallel but coordinated activities: operating costs and infrastructure costs. This paper focuses on the operating costs. The primary link between the two activities is the freight corridor. Particular corridor characteristics of importance include:

- Characteristics of the conventional freeway of the corridor:
    - homogeneous freeway segments: The default would be the segments corresponding to the truck-volume data. (See Appendix A.1.)
    - corresponding numbers of lanes on each of the segments (See Appendix A.1.)
    - average traffic volumes: both truck volume and volume of other vehicles (See Appendix A.1.)
- Characteristics of a truck-AHS constructed for the corridor:
    - The locations of AHS access and egress, in post miles from the border between California and Mexico (See Appendix A.2.)
- Characteristics of a truck lane constructed for the corridor:
    - The truck lane has the same configuration as the truck-AHS, including the same access points. (See Appendix A.2.)
    - The operational speed of the physically separated and dedicated truck lane, however, is lower than its truck-AHS counterpart.
    - Moreover, the threshold of trip distance beyond which a truck will use the truck lane is shorter than its truck-AHS counterpart. This is motivated by the absence of the mode-change requirement associated with a truck-AHS.

3.3 Performance Measures

Benefit and cost categories considered

Truck Operator Costs:

- Operating cost: Labor, Fuel
- Travel time

Non-truck Operator Costs:

- Travel time

Benefit and cost categories not considered in this study but considered in a companion study include the capital and maintenance costs and infrastructure costs, which in turn include the costs of the mainline, staging-area and other access and egress facilities.

3.4 Scenarios for Evaluation and Comparison

We address three aspects of the comparison scenarios: demand, truck-AHS configuration and truck-lane configuration, and assumptions about truck arrivals at and usage of the truck-AHS.

To study the performance of the three alternatives under future demand patterns, we inflate the current demand to 125% and 150%. We do not inflate the demand further because any of these higher demand levels would render at least one of the three alternatives inoperable, particularly the general-use portions of the alternatives.

Along the Interstate 5 within California, we selected 15 access points for a truck-AHS. The most southern access point is at the northern edge of San Diego while the most northern access point is at the border between California and Oregon. The average section length is approximately 50 miles. All the 15 locations can be found in Appendix A.2. The access points to the truck lane are identical to those to the truck-AHS.

In evaluation, we assume deterministic and uniformly distributed truck arrivals at an access point prior to using the AHS. In other words, the arrival times of trucks are equi-spaced but are commensurate with the demand pattern. For example, if 120 trucks go from one truck-AHS access point to another in one day, then a truck with this OD pattern will arrive at the origin access point every 5 minutes. The average wait is therefore half the headway.

A truck may or may not use the AHS. We assume that if the mainline portion of the trip of a truck exceeds PARAMETER - USAGE THRESHOLD, then it will use the AHS. Otherwise, it will not. We select 200 miles as the threshold value for a truck AHS.

Sometime, it is worthwhile for a truck to go backward away from the destination so that it can access the truck-AHS much more quickly than going forward toward the destination (to enter the truck-AHS). Similarly, sometimes it may be worthwhile to go past the destination freeway exit, leave the truck-AHS at the next AHS egress point, and then travel backward toward the destination freeway exit. The following parameter is used to decide whether a truck should go backward or not. If a truck will use the AHS (according to the previous rule) and the distance from the entry point of the truck on the freeway to the previous AHS access point divided by the total distance between the previous and the next AHS access points is less than BACK TRAVEL THRESHOLD, then the truck will travel to the previous AHS access point so as to access the AHS sooner. Otherwise, it will travel to the next AHS access point and then enter the truck-AHS there. Similarly, if the distance between the destination freeway exit of a truck and the first AHS egress point beyond that destination divided by the distance between that AHS egress point and the AHS egress point immediately before it is less than BACK TRAVEL THRESHOLD, the truck will travel to the farther AHS egress point first before traveling in reverse direction toward the destination freeway exit. We selected 0.1 for both thresholds.

We select 75 miles per hour as the operational speed for the truck-AHS.

Truck lane scenarios are identical to their truck-AHS counterparts except that the PARAMETER - USAGE THRESHOLD is set to be 50 miles and the operational speed is 60 miles per hour.

4. METHODOLOGY

This section describes the methodology for evaluating and comparing the three alternatives.

The overall performance measures of the truck-AHS or the truck-lane alternatives result from two related sets of measures estimating the performance of the system on two different components: (a) the conventional lanes and (b) truck-AHS or the truck lane. Given the knowledge of how the overall demand is split between the two components, the two sets of performance measures can be estimated. A key question is how to split the demand between the two components. Our approach is to first estimate the amount of truck traffic that will be attracted away from the conventional lanes onto the truck-AHS or the truck lane, and then, given the split traffic volumes, we solve two independent problems, one dealing with the conventional lanes while the other dealing with the truck-AHS or the truck lane. As mentioned earlier, we assume that the non-truck traffic varies with respect to the hour of the day but that truck traffic does not.

4.1 Performance of System on the Conventional Lanes

Since the conventional alternative of building a general-use lane involves only conventional lanes and both of the other two alternatives also involve conventional lanes, we first discuss how to estimate the performance of a conventional freeway, given the traffic demand for the conventional lanes. We will then discuss the estimation of performance measures related to the truck-AHS or the truck lane.

We consider only three levels of service and the three corresponding periods: AM/PM Peak, Near Peak and Free Flow. AM/PM Peak is assumed to be characterized by a flow rate of 2100 automobile-equivalents per hour at the speed of 35 miles per hour; Near Peak is assumed to be characterized by a flow rate of 1800 automobile-equivalents per hour at the speed of 50 miles per hour; Free Flow is assumed to be characterized by a flow rate of 1500 or below with a speed of 75 miles per hour. We first estimate the number of AM/PM Peak hours, and assume that the number of Near Peak hours is the maximum of 0 and half of the number of AM/PM Peak hours. The rest of the 24 hours, if any, are Free-Flow hours. For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or a

Near Peak hour is obtained by subtracting the average number of truck traveling the section per hour times 2 (to obtain the automobile-equivalents) from the flow rate of the corresponding hour. Based on these numbers, the three assumed speeds and the section lengths, both the total per-day truck travel time and total per-day non-truck travel time can be calculated.

This is how we estimate the number of AM/PM Peak hours in a day. Given daily section traffic counts, including the truck counts and the total counts, we first estimate the average number of automobile-equivalents per lane per hour (for one direction only but averaged over 24 hours) for each freeway section. Only two vehicle types are considered: trucks and non-trucks; one truck is considered as two automobile equivalents. We assume that if and only if, for any given section, the number of automobile-equivalents per lane per hour exceeds 500, then the section experiences peak hour congestion. For each additional 67 automobile-equivalents per lane per hour, the number of peak hours is increased by 1. If the number of automobile-equivalents per lane per hour is 2100 averaged over a 24-hour day, then this calculation produces 24 peak hours for the corresponding section. This is reasonable because in such a case, the demand of the section is so high that it is saturated all day. Such saturation occurs when the current demand is inflated to study the performance of the three alternatives in the future.

Truck traffic is assumed to be evenly distributed across the 24 hours of a day in a deterministic fashion. For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or a Near Peak hour is obtained by subtracting the average number of trucks traveling the section per hour times 2 (to obtain the automobile-equivalents) from the total flow rate of the corresponding hour. The total number of non-trucks that travel the section at the free flow can be obtained by subtracting the corresponding total numbers for the Peak and Near Peak hours from the total daily non-truck count. Based on these numbers, the three assumed speeds and the section lengths, both the total per-day truck travel time and total per-day non-truck travel time can be calculated.

Calculations for the fuel consumption and labor requirement for operations on the conventional lanes are straightforward. The labor requirement is exactly the same as the truck travel time. The fuel consumption is estimated based on the following average gas-mileages. 10 miles per gallon for trucks, regardless of speed.

4.2 Performance of System on the Truck-AHS or the Truck Lane

We now address the performance associated with the other component, i.e., the truck-AHS or the truck lane.

The problem of determining number of trucks that will be attracted away from the conventional lanes onto the truck-AHS or the truck lane is too big to be solved as one modeling-optimization problem. We use a simpler approach with the following characteristics. For each of the two alternatives (i.e., the truck lane or truck-AHS), our approach is similar. For the Truck-AHS option (or the truck-lane option), a truck operator is offered two choices: use AHS (or the truck lane) or not use AHS (or the truck lane). We summarize the approach using the context of truck-AHS. The approach is characterized by:

- Parameterized Decision Rules: The decision to be made by a trucking operator as to whether to use AHS or not is a very complex one, depending on the cost and travel-time advantages. We consider several decision rules, and each of the rules is parameterized at discrete levels.

- Estimate the recurring performance values resulting from the satisfaction of the OD demand for one day. (To get the recurring performance values for one year, for example, just multiply these values by 365.)

- Selection one or more most appropriate sets of the parameters as the optimal operational designs for truck-AHS and as the base for truck-AHS benefit-cost calculation. This can be enhanced in the future to include other important considerations, e.g., infrastructure costs and capital and maintenance costs of vehicles.

We now briefly describe our method for identifying the trucks that would use the truck-AHS lane and for estimating the relevant performance measures. Details can be found in Appendix B.

- **Use AHS or Not?** Determine the OD pairs for which the trucks will use the truck-AHS, using PARAMETER - AHS USAGE THRESHOLD. All trips longer than this threshold will use the truck-AHS. The OD trip numbers for trucks will be estimated by the Maximum-Entropy Method. Refer to Part I of this final report for details about the Maximum-Entropy Method. The threshold is set to be 200 miles. (The corresponding threshold for the truck lane alternative is 50 miles.)
- **Where to enter and exit?** If a truck uses the truck-AHS, determine if truck will travel backward or beyond to fully utilize the truck-AHS, using PARAMETER - BACK TRAVEL THRESHOLD. If a truck uses the truck-AHS and the distance between the conventional-freeway destination of a truck and the first AHS egress point beyond that destination divided by the distance between that AHS egress point and the AHS egress point immediately before it is less than BACK TRAVEL THRESHOLD, the truck will travel to the farther AHS egress point first before traveling in reverse direction toward the conventional freeway destination. The threshold is set to be 0.1. Same parameter and parameter values are used for determining the AHS access point used for entry. (These parameters and parameter values are also used for the truck-lane alternative.)
- **AHS Traffic by OD Per Day and the Resulting Conventional-Lane Traffic by Section Per day?** Given the (daily) OD trip numbers for the truck-AHS calculated in the previous two steps for all OD pairs, determine the (daily) truck counts for all the segments on the AHS. Deduct these counts from the (daily) section volume counts from the total demand and the remaining counts will be the traffic to travel on the conventional freeway. (This applies also to the truck-lane alternative.)
- **Truck Travel Time on AHS?** Determine the total travel time spent on the AHS by multiplying the travel time associated with a particular AHS OD and the corresponding OD counts and add a constant expected delay at the staging area that is equal to one half the PARAMETER - AHS TRACTOR HEADWAY plus AHS MODE CHANGE TIME. Assume that the AHS traffic moves at the design speed, which in our case is 75 miles per hour. AHS MODE CHANGE TIME is assumed to be 15 minutes. (For the truck-lane alternative, the operational speed for the truck lane is 60 miles per hour; no delay at the entrance is assumed for entry; no mode change is required either.) As mentioned earlier, although the end-to-end shuttle trucks may haul freight, the amount of freight to be hauled by these shuttle trucks depends on the freight demand. In this study, the freight demand is not sufficient to occupy all such shuttle trucks, and, we therefore, assume that none of such end-to-end shuttle trucks actually haul any freight. As a consequence, such shuttle trucks can be viewed as "overhead," and their use incurs additional travel time that the other two alternatives do not incur at all.

- **Labor Requirement for AHS Travel?** Determine the total per-day truck labor required to operate the truck-AHS. Only one driver is needed for one end-to-end shuttle convoy. (For the truck-lane alternative, the required labor time is simply the truck travel time.)
- **Fuel Requirement for AHS Travel**? Obtain the exact convoy size distribution for each section. Because of the constant headway, the convoy size for each of the convoys within each of the segments is constant and can be easily calculated. Calculate the average fuel requirement, with the reduction in fuel consumption fully considered. The fuel saving is set to be 10% for all the trailing trucks of a convoy. (No fuel saving is possible in the truck-lane alternative.)
- **Most Appropriate Parameter Set?** Select the most appropriate among the parameter sets studied. The selection would best be performed when infrastructure cost estimates are also considered. Because of our focus on the operating costs and on the software tools facilitating the overall assessment, we will obtain results for a set of parameters that we regard as the most appropriate.

We developed software tools to facilitate the study. The detailed requirements for the tools are specified in Appendix B.

5. SOFTWARE TOOLS

Five major software tools have been developed:

- Entropy Maximizer: maximum-entropy estimator for estimating the origins and destinations of the truck trips made in one day along the mainline freeway of a freight corridor, given only the daily volume by section
- Max-Entropy Problem Generator: problem specification tool that prepares the optimization problem for the maximum-entropy estimator
- Freeway Performance Estimator: performance estimator for the conventional-lanes portion of the system, including the entire system of the alternative of adding a general-use lane, the portion of conventional lanes of the truck-AHS alternative, and the portion of the conventional lanes of the truck-lane alternative.
- Truck-AHS Lane Performance Estimator: performance estimator for the truck-AHS-lane portion of the system, for the alternative of adding a truck-AHS lane
- Truck-Lane Performance Estimator: performance estimator for the truck-lane portion of the system, for the truck-lane alternative.

The Entropy Maximizer is identical to the one used in Part II. The reader is referred to Part II for details. The program listings are given in Appendix C.3 of Part II, and are not repeated in this Part IV.

The Max-Entropy Problem Generator is included in Appendix C.1. The Freeway Performance Estimator was developed using Excel. Program listings are not available, and hence are not included. However, it is relatively easy to create an identical Excel worksheet according to the earlier discussion or the specifications contained in the requirements document, which is provided in Appendix B. The program listings for the Truck-AHS Lane Performance Estimator are included in Appendix C.2. Since the Truck-Lane Performance Estimator is very similar to Truck-AHS Lane Performance Estimator, it is omitted.

6. NUMERICAL RESULTS

Based on the truck "link" counts summarized in Appendix A.1 for 160 sections of Interstate 5 within California and based on the entropy optimizer described in Section 5, we obtained maximum-entropy estimates of one-day truck trip numbers for all the possible pairs of 161 origins and 161 destinations, for three demand levels – 100%, 125% and 150% of the current demand. Due to the huge number of such pairs, we cannot provide all these trip numbers in this report. However, the trip length distributions corresponding to the three demand levels are summarized in Table 1. Table 1 also contains the numbers of truck trips that are attracted away from the conventional lanes to either the AHS lane or the conventional truck lane, under the truck-AHS or the truck-lane alternatives.

Table 1: Length Distributions of truck Trips

Demand: Percent of Current Level		100%	125%	150%
Truck Trip Length Distribution	0 - 50	30917	38685	46477
	50 - 100	9869	12361	14855
	100 - 150	3814	4781	5753
	150 - 200	2549	3194	3821
	200 - 250	1850	2316	2794
	250 - 300	1768	2202	2661
	300 - 350	948	1188	1428
	350 - 400	449	559	675
	400 - 450	215	267	326
	450 - 500	30	38	52
	500 - 550	18	25	33
	550 - 600	3	3	4
	600 - 650	3	5	6
	650 - 700	2	3	3
	700 - 750	0	0	0
	750 - 800	0	0	0
Total Number of Truck Trips		52435	65627	78888
# of Truck Trips Attracted to AHS		5286	6606	7982
# of Truck Trips Attracted to the Truck Lane		21518	26942	32411

The performance measures associated with the three alternatives are summarized in Tables 2, 3 and 4, for the demand levels of 100%, 125% and 150% of the current demand. These results reveal some very interesting phenomena. We address these phenomena in the order of total travel time (including trucks and non-trucks), total truck travel time, truck labor requirement, truck fuel requirement.

Table 2: Performance Measures of the Three Alternatives: 100% of Current Demand

ALTERNATIVES		GENERAL-USE LANE	TRUCK-AHS	TRUCK LANE
NON-CONV PORTION (AHS OR TRUCK LANE)	TRUCK TIME	0	1186413	249
	TRUCK LABOR	0	87364	249
	TRUCK FUEL	0	121654	24
CONV PORTION	TRUCK TIME	3262635	2286360	108
	NON-TRUCK TIME	30839720	33070548	3214
	TRUCK LABOR	3262635	2286360	108
	TRUCK FUEL	359999	225234	11
TRUCK TOTALS	TOTAL TRUCK TIME	3262635	3472773	358
	TOTAL TRUCK LABOR	3262635	2373724	358
	TOTAL TRUCK FUEL	359999	346888	36
NON-TRUCK TRAVEL TIME		30839720	33070548	3214
TOTAL TRAVEL TIME		34102354	36543321	3573

Units: Travel time and labor requirement are in minutes; fuel requirements are in gallons.

Table 3: Performance Measures of the Three Alternatives: 125% of Current Demand

ALTERNATIVES		GENERAL-USE LANE	TRUCK-AHS	TRUCK LANE
NON-CONV PORTION (AHS OR TRUCK LANE)	TRUCK TIME	0	1476189	312
	TRUCK LABOR	0	109205	312
	TRUCK FUEL	0	152027	31
CONV PORTION	TRUCK TIME	4368011	3119929	148
	NON-TRUCK TIME	41697748	43499871	4272
	TRUCK LABOR	4368011	3119929	148
	TRUCK FUEL	449998	281566	13
TRUCK TOTALS	TOTAL TRUCK TIME	4368011	4596118	460
	TOTAL TRUCK LABOR	4368011	3229134	460
	TOTAL TRUCK FUEL	449998	433593	45
NON-TRUCK TRAVEL TIME		41697748	43499871	4272
TOTAL TRAVEL TIME		46065759	48095989	4733

Units: Travel time and labor requirement are in minutes; fuel requirements are in gallons.

Table 4: Performance Measures of the Three Alternatives: 150% of Current Demand

ALTERNATIVES		GENERAL-USE LANE	TRUCK-AHS	TRUCK LANE
NON-CONV PORTION (AHS OR TRUCK LANE)	TRUCK TIME	0	1779490	3761
	TRUCK LABOR	0	131046	3761
	TRUCK FUEL	0	183809	376
CONV PORTION	TRUCK TIME	5541239	4027553	1901
	NON-TRUCK TIME	52091527	53985840	53181
	TRUCK LABOR	5541239	4027553	1901
	TRUCK FUEL	539998	336336	164
TRUCK TOTALS	TOTAL TRUCK TIME	5541239	5807043	5663
	TOTAL TRUCK LABOR	5541239	4158599	5663
	TOTAL TRUCK FUEL	539998	520145	541
NON-TRUCK TRAVEL TIME		52091527	53985840	53181
TOTAL TRAVEL TIME		57632765	59792883	58844

Units: Travel time and labor requirement are in minutes; fuel requirements are in gallons.

For ease of discussion, we refer to the three alternatives as

- General-Use Lane (adding one conventional general-use lane per direction to the conventional freeway)
- Truck-AHS (constructing a physically separated one-lane truck-AHS within or along the right-of-way of a conventional freeway)
- Truck Lane (constructing a physically separated lane dedicated to truck travel within or along the right-of-way of a conventional freeway).

In terms of the total travel time (including both the total truck travel time and its non-truck counterpart), a General-use Lane is better than a Truck-AHS and is also better than a Truck Lane; this is true for all the three demand levels considered. This is because the conventional freeway lanes of either the truck-AHS or the Truck Lane alternatives are much more congested than their General-Lane counterpart. In fact, the truck-AHS has a higher Total Travel Time than the Truck Lane. This is because the Truck Lane attracts many more trucks away from the conventional freeway lanes leaving the conventional lanes somewhat less congested, despite the fact that the assumed speed of 60 miles per hours for a Truck Lane is higher than the assumed speed of 75 miles per hour for the truck-AHS. Finally, these differences are not very significant. The percentages of the differences with respect to the highest amounts of total travel time are 6.7%, 4.2% and 3.6% for the demand levels of 100%, 125% and 150%, respectively. The difference in terms of percentage decreases.

One might expect that at least the Truck-AHS and the Truck-Lane alternatives would have an advantage in the total truck travel time. However, even this is not true; the General-Use Lane alternative is better than both the Truck-AHS and Truck-Lane alternatives in total truck travel time. This is because those trucks whose trip lengths do not warrant the use of either the truck-AHS or the Truck-Lane (for the corresponding alternative) will have to travel with the non-truck traffic on the much more congested conventional freeway lanes. At the 100% and 125% demand levels, Truck-AHS alternative is better than the Truck-Lane alternative while at the 150%

demand level, the reverse is true. The shorter trip distance requirement for use of the Truck-Lane alternative leads to a much higher usage than its Truck-AHS counterpart and also leads to the total-truck-travel-time advantage at the 150% demand level. The percentages of the differences with respect to the highest amounts of total truck travel time are 9.0%, 5.1% and 4.6% for the demand levels of 100%, 125% and 150%, respectively. The difference in terms of percentage again decreases.

In terms of truck-labor requirements, the Truck-AHS is clearly the best one while the Truck Lane is the worst. This is consistent regardless of the difference in the three demand levels. The percentages of the differences with respect to the highest amounts of total truck-labor requirements are 34%, 30% and 27% for the demand levels of 100%, 125% and 150%, respectively. The difference in terms of percentage decreases, however.

In terms of truck fuel requirements, Truck-AHS is better than General-Use Lane, which is better than Truck Lane. This is true regardless of the difference in the three demand levels. The percentages of the differences with respect to the highest amounts of total truck fuel requirement is approximately 4%.

Based on these results, it is clear that, under the assumptions made and given the corridor selected, the Truck-AHS alternative will not provide any travel time advantage, except to the long-haul trucking industry at the expense of the short-haul trucking industry and the non-truck driving public. (Trucks using the truck-AHS travel at a consistently higher speed of 75 miles per hour.)

The only advantages of truck-AHS are labor and fuel savings. These savings must be weighed against the infrastructure costs. Moreover, the safety and technical feasibility of the truck-AHS must be carefully studied. The Truck Lane alternative does not look promising either; it does not enjoy any advantage, and we have not even begun to address the cost of infrastructure.

Finally, for the General-Use Lane alternative, we compared the addition of only one conventional general-use lane to the other two alternatives. After the “overhead” infrastructure requirements, e.g., the break-down lane or shoulder and the width required for physical barriers, etc., for the other two alternatives are taken into consideration, it is likely that the overall right-of-way required by either of the other two alternatives can accommodate two conventional general-use lanes. As a result, the travel time advantages of the General-Use Lane alternative just reported will likely be clear understatements. In addition, addition of two general-use lanes, without the need for physical separation as required by the other two alternatives, will likely cost drastically less.

Our comparison suggests that general-use lane is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings.

Although these results can provide valuable insights into the advantages and disadvantages of the three alternatives in general, the reader is reminded that what we have evaluated and compared are three specific operational systems and that the comparison is made against a specific reference corridor. In addition, due to the complexity of the problem and the absence of daily origin-destination data for truck trips and time-dependent demand data for freeway sections, several estimation methods have been employed. Caution is needed when generalizing these specific results to the three corresponding general alternatives. We now summarize possible limitations as follows.

The truck-AHS evaluated and compared is a closed system. Although this closed system can bypass several major deployment issues, it requires a shuttle AHS tractor to lead any convoy traveling on the AHS. This shuttle AHS tractor may or may not have its own freight to haul. We assume that such AHS tractors serve only as a "guide," without any freight of its own to haul. This assumption is based on the origin-destination trip numbers estimated by the entropy-maximization method for the reference corridor; the number of trucks traveling from one end to the other on the corridor is negligible. (See Table 1 above.) Therefore, these AHS tractors can be viewed as "overhead," and their travel time is included in the total truck travel time. Such AHS tractors are not required for open AHS systems. However, deployment of such open systems may be more difficult or may require a much longer timeframe. In addition, their operations may also be much less organized and may require much more coordination among the users. Regarding the Truck-Lane alternative, we selected 60 miles as the traffic speed because only one lane is provided, the speed of traffic is limited by that of the slowest truck, and it is not realistic to forbid trucks capable of a 60-mile cruising speed from using the truck lane.

The origins and destinations of the truck trips were needed to determine which truck trips would use the AHS or the truck lane. Their unavailability of such data necessitated estimation. We used the maximum-entropy method to estimate the daily "OD numbers" for truck trips. These estimates may not be accurate. For cases where such data are available, the tools developed for this research can be used to provide more accurate results.

The reference corridor is the California portion of Interstate 5. The length of this corridor is approximately 770 miles. The longer the corridor length, the higher benefit that can be realized by the provision of a truck AHS. This reference corridor may not be long enough. Although the use of a longer corridor may result in more travel-time benefit, the use of this corridor provides a glimpse into the realism that an extensive truck AHS would be built gradually and a subsystem of smaller scale alone should also be able to provide sufficient benefit to justify the construction. In the absence of a mode-choice model regarding the use of AHS or the use of truck lane, we selected 200 miles as the threshold for AHS use and 50 miles for truck-lane use. (The corresponding threshold for rail is commonly perceived as 500 miles or longer.) Our AHS and truck-lane benefit estimates may be quite optimistic because we assume that all truck trips longer than these thresholds would use the corresponding facilities.

The routing for the AHS operations is simple. There is only one route – the end-to-end shuttle. This simplicity leads to inefficient operations. For example, a shuttle may lead a convoy of size 3 or 4. Partial routes can be added to improve the efficiency. However, operating partial routes requires an additional set of operating rules that must be supported by additional vehicle-infrastructure capabilities. The scheduling is also simple; it is based on a constant headway. For cases where time-dependent truck demand data are available, more efficient scheduling can be developed to achieve higher efficiency.

Based on our results and given these limitations, we believe that future studies on truck-AHS should be focused on the feasibility of driverless, automated, closely-spaced truck-following as potential source of labor and fuel savings and on the concomitant infrastructure costs if the purpose of constructing a truck-AHS is to facilitate inter-city trucking. Other truck-AHS operating concepts may be developed for and may benefit other more special purposes.

7. CONCLUDING REMARKS

This research developed a new truck-AHS operating concept and provided sufficient operational details for a quantitative evaluation and comparison with two conventional alternatives. The new concept was motivated to respond to user needs and to increase the deployability while taking advantage of promising technologies. We compared the following three alternatives:

- General-Use Lane (adding one conventional general-use lane per direction to the conventional freeway)
- Truck-AHS (constructing a physically separated one-lane truck-AHS within or along the right-of-way of a conventional freeway)
- Truck Lane (constructing a physically separated lane dedicated to truck travel within or along the right-of-way of a conventional freeway).

Based on our numerical results, it is clear that, under the assumptions made and given the corridor selected, the Truck-AHS alternative will not provide any travel time advantage to the overall system or even to the overall trucking industry. It is able to provide travel time advantage to only the long-haul trucking industry but at the expense of the short-haul trucking industry and the non-truck driving public. (Trucks using the truck-AHS travel at a consistently higher speed of 75 miles per hour.)

The only advantages of truck-AHS are labor and fuel savings, and the labor saving is drastic. These savings must be weighed against the infrastructure costs. Moreover, the safety and technical feasibility of the truck-AHS must be carefully studied. The Truck Lane alternative does not look promising either; it does not enjoy any advantage over the General-Use Lane alternative, and we have not even begun to address the cost of infrastructure.

Finally, for the General-Use Lane alternative, we compared the addition of only one conventional general-use lane to the other two alternatives. After the "overhead" infrastructure requirements, e.g., the break-down lane or shoulder and the width required for physical barriers, etc., for the other two alternatives are taken into consideration, it is likely that the overall right-of-way required by either of the other two alternatives can accommodate two conventional general-use lanes. As a result, the travel time advantages of the General-Use Lane alternative just reported will likely be clear understatements. In addition, addition of two general-use lanes, without the need for physical separation as required by the other two alternatives, will likely cost drastically less.

Our comparison suggests that general-use lane is a clear winner of the three alternatives unless the driverless, automated, closely-spaced truck-following can be implemented safely so as to reap the possible labor and fuel savings.

Although these results can provide valuable insights into the advantages and disadvantages of the three alternatives in general, the reader is reminded that what we have evaluated and compared are three specific operational systems and that the comparison is made against a specific reference corridor. In addition, due to the complexity of the problem and the absence of daily origin-destination data for truck trips and time-dependent demand data for freeway sections, several estimation methods have been employed. Caution is needed when generalizing these specific results to the three corresponding general alternatives.

Based on our results and given these limitations, we believe that future studies on truck-AHS should be focused on the feasibility of driverless, automated, closely-spaced truck-following as potential source of labor and fuel savings and on the concomitant infrastructure costs if the purpose of constructing a truck-AHS is to facilitate inter-city trucking. Other truck-AHS operating concepts may benefit other more special purposes.

REFERENCES

Al-Ayat, R., and Hall, R.W., "A conceptual approach for developing and analyzing alternate evolutionary deployment strategies for Intelligent Vehicle/Highway Systems," PATH Working Paper 94-5, 1994.

AlKadri, M., Benouar, H., and Tsao, H.-S.J., "Intermediate Automation Concepts for Evolution toward Automated Highway Systems," Transportation Research Record No. 1651, 1998.
Bagby, J.W., and Gittings, G.L., "Litigation Risk Management for Intelligent Transportation Systems," ITS Quarterly, Spring-Summer, 1999.

Baum, H. and Schulz, W.H., "Economic Evaluation of CHAUFFEUR," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Borodani, P., Carrea, P., and Gortan, L., "Short headway Control of Trucks on Motorways: the CHAUFFEUR Project," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Calspan Corporation, Precursor Systems Analyses of Automated Highway Systems Commercial and Transit AHS Analysis. Vol. VII, FHWA-RD-95-134, Federal Highway Administration, U.S. Department of Transportation, 1995.

Fang, S.-C. and Tsao, H.-S.J., (1995) "Linearly-Constrained Entropy Maximization Problem with Quadratic Costs and Its Applications to Transportation Planning Problems," Transportation Science, Vol. 29, No.4, pp. 353-365, 1995.

Fang, S.-C., Tsao, H.-S.J., and Rajasekera, J.R., (1997) Entropy Optimization And Mathematical Programming, Kluwer Academic Publishers, September, 1997.

Hall, R.W., "System Configurations: Evolutionary Deployment Considerations," Chapter 4 of Automated Highway Systems, P.A. Ioannou (editor), Plenum Press, New York, pp. 49-71, 1997.

Hall, R.W., Nowroozi, A., and Tsao, H.-S.J., "Entrance Capacity of an Automated Highway System," to *Transportation Science*, Vol. 35, No. 1, pp. 19 – 36, 2001.

Hall, R.W. and Tsao (1997a), H.-S. J. "Automated Highway System Deployment: A Preliminary Assessment of Uncertainties", Chapter 16 of Automated Highway Systems, P.A. Ioannou (editor), Plenum Press, New York, pp. 325 - 334, 1997.

Hall, R.W., and Tsao, H.-S. J. (1997b), "Capacity of Automated Highway Systems: Merging Efficiency," Proceedings of American Control Conference, Albuquerque, New Mexico, June, 1997, pp. 2046-2050.

Riva, P. and Ulken, U., "CHAUFFEUR Tow-Bar: the Vehicle Based System Architecture," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Schulze, M., "CHAUFFEUR, the European Way towards an Automated Highway System," Proceedings of the 4th World Congress on Intelligent Transportation Systems, Berlin, Germany, 1997.

Sengupta, R., Godbole, D. and H.-S.J. Tsao, "Concept Definition of an Infrastructure-Supported Automated Highway System," Proceedings of The Third World Congress on Intelligent Transportation Systems, Orlando, Florida,1996.

Shladover, S.E., "Progressive Deployment Steps Leading Toward an Automated Highway System," Transportation Research Record 1727, pp. 154 - 161, Transportation Research Board, 2000.

Tai, Meihua, Advanced vehicle control of heavy vehicles for automated highway systems, Ph.D. Thesis, Department of Mechanical Engineering, University of California, Berkeley, 2001.

Tai, Meihua; Wang, Jeng-Yu; White, Ryan; Tomizuka, Masayoshi, Robust lateral control of heavy duty vehicles, PATH research report ; UCB-ITS-PRR-2001-35, University of California, Berkeley. Institute of Transportation Studies Partners for Advanced Transit and Highways (Calif.) University of California, Berkeley, Dec. 2001.

Tsao, H.-S. J. (1995a), "Traffic Control for Automated Highway Systems: A Conceptual Framework", Transportation Research, Part C, Vol. 3, No. 4, pp. 227-246, 1995.

Tsao, H.-S. J. (1995b), "Stage Definition for AHS Deployment and an AHS Evolutionary Scenario", Intelligent Vehicle Highway Systems Journal, Vol.2(4), pp. 359-382, 1995.

Tsao, H.-S. J. (1995c), "A Staggered-Diamond Design for Automated/Manual-HOV Highway-to-Highway Interchanges and Constraints on AHS Design for Accommodating Automated Highway Change", Intelligent Vehicle Highway Systems Journal, Vol.2(3), pp. 281-292, 1995.

Tsao, H.-S. J. (1995d), "Constraints on Initial AHS Deployment and the Concept Definition of a Shuttle Service for AHS Debut", IVHS Journal, Vol. 2(2), 159-173, 1995.

Tsao, H.-S. J. (1998a), "From the Guest Editor," Special Issue of the ITS Journal on Automated Highway Systems, Vol. 4, Nos:1-2, 1998.

Tsao, H.-S.J. (1998b), "An Axiomatic Approach to Developing Partial Automation Concepts for Deployment of Automated Highway Systems and Partial Invocation of Vision-Based Lane-Keeping and Adaptive Cruise Control," *Transportation Research Record*, No. 1651, 1998.

Tsao, H.-S. J. (2001), " A Framework for Evaluating Deployment Strategies for Intelligent Transportation Systems", Intelligent Transportation Systems Journal (ITS Journal), Vol. 6, pp. 141-173, 2001.

Tsao, H.-S.J. and Botha, J.L. (2001), "Operating Concepts for Urban Bus Automation and Inter-city Truck Automation," the Proceedings of the 2001 IEEE ITS Conference, Oakland, California.

Tsao, H.-S.J. and Botha, J.L (2002a) , "An Automated Highway System Dedicated To Inter-City Trucking: Operating Concepts And Deployment," presented at the 2002 Annual Meeting of the Transportation Research Board on Jan. 14, 2002, Washington, D.C.

Tsao, H.-S.J. and Botha, J.L. (2002b), "An Automated Highway System Dedicated To Inter-City Trucking: Design Options, Operating Concepts And Deployment," Intelligent Transportation Systems Journal (The ITS Journal), Vol. 7, No. 2, pp. 169 – 196, 2002.

Tsao, H.-S. J., Botha, J.L. (2003), "Definition and Evaluation of Bus and Truck Automation Operations Concepts: Final Report," California PATH Research Report UCB-ITS-PRR-2003-19, May 2003.

Tsao, H.-S. J. and Hall, R.W., "A Probabilistic Model for AVCS Longitudinal Collision/Safety Analysis", Intelligent Vehicle Highway Systems Journal, Vol.1, No. 3, pp. 261-274, 1994.

Tsao, H.-S. J., Hall, R.W., and Chatterjee, I., "Analytical Models for Vehicle/Gap Distribution on Automated Highway Systems", Transportation Science, Vol. 31, No. 1., pp. 18-33, 1997.

Tsao, H.-S. J., R.W. Hall and S.E.Shladover, "Design Options for Operating Automated Highway Systems", Proceedings of Vehicle Navigation & Information Systems Conference (Oct. 1993).

Ulmer, Berthold, "CHAUFFEUR: From Concept to Realization," 3rd International Workshop on Vehicle-Highway Automation, Toronto, Canada, Nov. 11-12, 1999.

APPENDIX A: THE REFERENCE FREIGHT CORRIDOR

APPENDIX A.1: Idealized Interstate 5 Homogeneous Freeway Segments in California

Location #	Location	Miles to Border	Section Length	Idealized # Lanes	ADT (2 directions)	Truck ADT (2 directions)	Truck %
1	SAN DIEGO, MEXICAN	0.09	0.09	6	98000	2156	2.2
2	SOUTH JCT. RTE. 805	0.88	0.79	6	45000	1710	3.8
4	JCT. RTE. 75 WEST	4.63	3.75	4	152000	5624	3.7
5	8TH STREET	11.13	6.50	4	184000	9200	5.0
7	JCT. RTE. 15 NORTH	12.65	1.52	4	154000	6314	4.1
9	SAN DIEGO, JCT. RTE.	14.08	1.43	4	179000	7160	4.0
11	SAN DIEGO, JCT. RTE.	15.04	0.96	4	177000	7080	4.0
13	SAN DIEGO, JCT. RTE.	16.07	1.03	4	203000	8323	4.1
15	JCT. RTE. 8/ROSECRANS	20.06	3.99	4	204000	6936	3.4
17	SAN DIEGO, JCT. RTE.	23.48	3.42	4	143000	5720	4.0
18	JCT. RTE. 52 EAST	25.95	2.47	4	184000	7544	4.1
20	SAN DIEGO, NORTH JCT.	30.68	4.74	4	261000	9918	3.8
22	JCT. RTE. 78 EAST	51.20	20.52	4	211000	10550	5.0
23	JCT. RTE. 76 EAST	53.43	2.23	4	163000	10106	6.2
24	BASILONE ROAD	71.38	17.95	4	139000	10842	7.8
25	SAN DIEGO/ORANGE	73.00	1.62		134000	9675	7.2
26	JCT RTE 1	79.78	6.78	4	229000	9733	4.3
28	SAN JUAN CAPISTRANO,	82.60	2.82	4	241000	9592	4.0
29	CROWN VALLEY PARKWAY	86.78	4.17	4	280000	9800	3.5
30	JCT. RTE. 405, SANTA	94.30	7.53	4	285000	9605	3.4
32	IRVINE, JCT. RTE. 133	96.12	1.82	4	235000	14688	6.3
34	TUSTIN, JCT. RTE. 55,	103.26	7.14	4	262000	16768	6.4
36	SANTA ANA, JCT. RTES.	107.00	3.74	4	234000	16380	7.0
37	ORANGE, CHAPMAN AVE	107.94	0.94	3	224000	15680	7.0
39	KATELLA AVENUE	109.26	1.32	3	214000	20544	9.6
41	LINCOLN AVENUE	111.92	2.66	3	232000	22272	9.6
43	FULLERTON, JCT. RTE.	115.10	3.18	3	170000	15980	9.4
44	JCT. RTE. 39; BEACH	116.43	1.33	3	170000	18020	10.6
45	ORANGE/ LOS ANGELES	118.00	1.57		177000	17700	10.0
47	SANTA FE SPRINGS, JCT.	124.85	6.85	4	230000	16560	7.2
49	COMMERCE, JCT. RTE.	131.78	6.94	4	256000	20122	7.9
50	ESPERANZA STREET	133.33	1.55	4	263000	20330	7.7
52	LOS ANGELES, JCT. RTE.	134.47	1.14	4	241000	19497	8.1
54	LOS ANGELES, JCT. RTE.	136.45	1.98	4	243000	18347	7.6
56	LOS ANGELES, JCT. RTE.	138.44	1.99	4	281000	18153	6.5

58	LOS ANGELES, JCT. RTE.	140.55	2.10	4	248000	17757	7.2
59	LOS ANGELES, COLORADO	143.78	3.23	4	264000	17952	6.8
61	LOS ANGELES, JCT. RTE.	145.08	1.30	4	221000	17592	8.0
63	SUN VALLEY, JCT. RTE.	154.36	9.28	4	280000	21812	7.8
64	JCT. RTE. 118,	157.36	3.00	5	260000	21814	8.4
66	SYLMAR, JCT. RTE. 405	159.60	2.24	4	250000	16550	6.6
68	TUNNEL STATION, JCT.	160.39	0.79	5	184000	17554	9.5
69	SANTA CLARITA, SOUTH	168.37	7.98	4	144000	17208	12.0
71	NORTH JCT. RTE. 126	170.29	1.91	4	88000	16060	18.3
72	SOUTH JCT. RTE. 138,	196.29	26.01	4	60000	16374	27.3
73	NORTH JCT. RTE. 138	196.91	0.62	4	62000	17000	27.4
75	LOS ANGELES/KERN	203.41	6.50		61000	17001	27.9
77	JCT. RTE. 99 NORTH	219.27	15.86	4	28000	8120	29.0
78	JCT. RTE. 166	223.02	3.75	2	28000	8120	29.0
80	JCT. RTE. 119	242.21	19.18	2	28500	8265	29.0
82	JCT. RTE. 43	244.61	2.40	2	28000	7840	28.0
84	JCT. RTE. 58	255.56	10.95	2	30000	9000	30.0
86	JCT. RTE. 46	276.43	20.87	2	29000	8990	31.0
88	JCT. RTE. 41	306.42	30.00	2	28500	8550	30.0
90	JCT. RTE. 198	331.30	24.87	2	28500	8550	30.0
92	JCT. RTE. 33 SOUTH,	334.39	3.09	2	29500	8850	30.0
94	JCT. RTE. 165 NORTH	375.67	41.28	2	30500	8479	27.8
95	JCT. RTE. 152	386.97	11.30	2	28000	9013	32.2
96	JCT. RTE. 33	391.23	4.26	2	30500	8479	27.8
97	JCT. RTE. 140 EAST	401.78	10.55	2	32000	8896	27.8
99	JCT. RTE. 580 WEST	402.46	0.68	2	14300	3961	27.7
100	JCT. RTE. 132	405.22	2.76	2	19600	5449	27.8
101	JCT. RTE. 33 SOUTH	408.25	3.02	2	18700	6122	32.7
102	OLD ROUTE 50; 11TH	413.58	5.33	2	19300	5192	26.9
103	JCT. RTE. 205 WEST	414.40	0.82	3	127000	33528	26.4
104	JCT. RTE. 120 EAST	416.61	2.21	3	70000	18130	25.9
106	FRENCH CAMP	422.73	6.12	3	65000	16250	25.0
108	STOCKTON, JCT. RTE. 4	427.14	4.41	3	101000	24240	24.0
109	STOCKTON, JCT. RTE. 4	427.96	0.82	3	108000	25380	23.5
110	MARCH LANE	431.77	3.81	3	94000	21620	23.0
111	STOCKTON, HAMMER LANE	434.44	2.67	3	63000	14238	22.6
113	JCT. RTE. 12	441.35	6.91	3	49500	10544	21.3
114	WALNUT GROVE ROAD	449.38	8.03	2	48000	12202	25.4
115	SAN JOAQUIN/SACRAMENTO	452.40	3.02	2	47000	11947	25.4
116	LAMBERT ROAD	457.03	4.63	2	49000	11946	24.4
117	SACRAMENTO,	468.53	11.50	2	92000	12880	14.0
118	SACRAMENTO, JCT. RTE.	474.95	6.42	4	129000	12384	9.6
119	SACRAMENTO, I STREET	476.18	1.23	4	157000	15072	9.6
121	SACRAMENTO, JCT.	479.10	2.92	4	98000	12407	12.7

	RTE.						
123	SACRAMENTO, JCT. RTE.	482.29	3.19	4	70000	13650	19.5
125	WOODLAND, EAST MAIN	494.80	12.51	2	34000	7820	23.0
127	JCT. RTE. 505 SOUTH	510.90	16.10	2	30000	8400	28.0
129	JCT. RTE. 20	535.62	24.72	2	25000	5740	23.0
131	WILLOWS, JCT. RTE. 162	560.99	25.37	2	21600	5739	26.6
133	JCT. RTE. 32 EAST	576.65	15.66	2	20000	5820	29.1
135	GLENN/TEHAMA COUNTY	579.94	3.29		22600	5910	26.2
136	LIBERAL AVENUE	585.71	5.77	2	22500	5850	26.0
137	SOUTH AVENUE	587.79	2.08	2	24900	5926	23.8
138	CORNING ROAD	588.92	1.13	2	26500	5925	22.4
139	FINNELL AVENUE	590.91	1.99	2	25500	6120	24.0
140	GYLE ROAD	593.91	3.00	2	24200	6111	25.3
141	FLORES AVENUE	599.72	5.82	2	24700	6052	24.5
142	RED BLUFF, SOUTH MAIN	604.81	5.09	2	26000	6500	25.0
143	RED BLUFF, DIAMOND	604.88	0.07	2	28000	6972	24.9
144	RED BLUFF, JCT. RTE.	606.47	1.58	2	29000	7323	25.3
145	NORTH RED BLUFF	608.32	1.85	2	36500	7483	20.5
146	WILCOX ROAD	610.98	2.67	2	35500	7459	21.0
147	JELLYS FERRY ROAD	612.18	1.19	2	34000	7456	21.9
148	HOOKER CREEK ROAD	616.31	4.13	2	33500	7454	22.3
149	SNIVELY ROAD	618.66	2.35	2	33500	7454	22.3
150	BOWMAN ROAD	621.47	2.81	2	34000	7160	21.1
153	FOURTH STREET	622.97	1.50	2	40000	7752	19.4
155	JCT. RTE. 273 NORTH	625.89	2.92	2	37500	6844	18.3
157	ANDERSON, BALLS FERRY	627.35	1.46	2	36500	7548	20.7
158	ANDERSON, NORTH STREET	627.70	0.35	2	44500	7725	17.4
159	RIVERSIDE AVENUE	628.80	1.10	2	45500	7589	16.7
160	KNIGHTON ROAD	631.83	3.03	2	47500	7477	15.7
161	CHURN CREEK ROAD	634.21	2.38	2	51000	7676	15.1
162	CYPRESS STREET	636.52	2.31	2	60000	7560	12.6
163	REDDING, JCT. RTE. 299	637.50	0.99	2	43000	5951	13.8
164	REDDING, JCT. RTE. 299	639.38	1.87	2	44500	6230	14.0
165	REDDING, TWINVIEW	640.12	0.75	2	37500	5700	15.2
166	REDDING, JCT. RTE. 273	640.54	0.41	2	43500	5738	13.2
167	REDDING, OASIS ROAD	641.46	0.92	2	34500	5644	16.4
168	PINE GROVE	643.05	1.59	2	31500	5639	17.9
169	JCT. RTE. 151 WEST	644.20	1.15	2	22500	5776	25.7
170	MOUNTAIN GATE	646.14	1.94	2	20300	5751	28.3
171	FAWNDALE	648.09	1.95	2	19500	6445	33.1
172	BRIDGE BAY	649.69	1.60	2	18800	5713	30.4
173	TURNTABLE BAY ROAD	651.37	1.68	2	18800	5713	30.4
174	O' BRIEN	654.22	2.84	2	17900	2855	16.0
175	GILMAN ROAD	658.88	4.67	2	17500	5686	32.5
176	ANTLER	663.11	4.23	2	17000	5651	33.2

177	LAKEHEAD	664.37	1.26	2	16900	5650	33.4
178	GIBSON	674.96	10.58	2	16700	5648	33.8
179	SIMS ROAD	679.47	4.51	2	16700	5648	33.8
180	FLUME CREEK	681.41	1.94	2	16700	5648	33.8
181	CONANT ROAD	682.56	1.16	2	16700	5648	33.8
182	SWEETBRIER AVENUE	683.80	1.24	2	16700	5650	33.8
183	CASTELLA	685.64	1.84	2	16800	5665	33.7
184	SODA CREEK ROAD	687.47	1.83	2	16900	5670	33.6
185	CRAIG VIEW DRIVE	688.06	0.59	2	17200	5671	33.0
186	CASTLE CRAGS DRIVE	688.90	0.84	2	17200	5998	34.9
188	SHASTA/SISKIYOU COUNTY	689.08	0.18	2	17200	5685	33.1
189	SOUTH DUNSMUIR	689.76	0.68	2	16300	5654	34.7
190	CENTRAL DUNSMUIR	691.59	1.83	2	17500	5745	32.8
191	DUNSMUIR, DUNSMUIR	692.92	1.33	2	18200	5695	31.3
192	MOTT AVENUE	694.97	2.06	2	18800	5715	30.4
193	JCT. RTE. 89 EAST	697.55	2.58	2	18300	5560	30.4
194	MOUNT SHASTA, LAKE	699.56	2.01	2	18200	5294	29.1
195	NORTH MOUNT SHASTA	701.14	1.58	2	22200	5450	24.6
196	ABRAMS LAKE ROAD	702.26	1.12	2	21300	5112	24.0
197	DEETZ ROAD	704.41	2.16	2	21500	5182	24.1
198	SOUTH WEED	706.52	2.10	2	20500	4941	24.1
199	JCT. RTE. 97 NORTH	708.15	1.63	2	14000	4025	28.8
200	JCT. RTE. 265	708.93	0.79	2	15600	3900	25.0
201	EDGEWOOD	712.07	3.14	2	14700	3812	25.9
202	WEED AIRPORT	714.42	2.35	2	14800	3812	25.8
203	LOUIE ROAD	720.25	5.83	2	14800	3812	25.8
204	GRENADA	727.28	7.03	2	16200	4160	25.7
205	KILLGORE HILLS ROAD	731.58	4.30	2	16600	4175	25.2
206	SOUTH YREKA	734.70	3.11	2	15000	3821	25.5
207	YREKA, MINER STREET	736.64	1.94	2	13700	3820	27.9
208	YREKA, JCT. RTE. 3	737.31	0.68	2	13200	3740	28.3
209	JCT. RTE. 96 WEST	747.40	10.09	2	13700	3600	26.3
210	HENLEY WAY	750.63	3.23	2	13700	3574	26.1
211	DITCH CREEK ROAD	752.00	1.37	2	13700	3561	26.0
212	BAILEY HILL ROAD	754.59	2.60	2	13700	3557	26.0
213	HILT ROAD	757.40	2.81	2	13700	3713	27.1
214	OREGON STATE LINE	758.37	0.97	2	13700	3546	25.9

APPENDIX A.2: HYPOTHESIZED TRUCK AHS ACCESS POINTS ON I-5 IN CA

DISTRICT	COUNTY	LOCATION	POST MILE
11	SD	SAN DIEGO, NORTH JCT.	30.68
12	ORA	IRVINE, JCT. RTE. 133	96.12
7	LA	COMMERCE, JCT. RTE.	131.78
7	LA	JCT. RTE. 118,	157.36
7	LA	LOS ANGELES/KERN	203.41
6	KER	JCT. RTE. 46	276.43
6	KIN	JCT. RTE. 41	306.42
6	FRE	JCT. RTE. 33 SOUTH,	334.39
10	MER	JCT. RTE. 152	386.97
10	SJ	JCT. RTE. 120 EAST	416.61
3	SAC	SACRAMENTO, JCT. RTE.	474.95
3	GLE	WILLOWS, JCT. RTE. 162	560.99
2	SHA	REDDING, JCT. RTE. 273	640.54
2	SIS	JCT. RTE. 89 EAST	697.55
2	SIS	OREGON STATE LINE	758.37

APPENDIX B: REQUIREMENTS FOR "SOFTWARE TOOLS FOR EVALUATING AND COMPARING TRUCKING ALTERNATIVES"

SCOPE OF EVALUTION AND COMPARISON

Alternatives

Evaluate and compare the benefit and cost of

- Adding an exclusive AHS truck lane

to the those of the following two conventional alternatives:

- Adding a dedicated truck lane
- Adding a conventional lane (without dedication of any lanes to truck use).

Benefit and cost categories considered

Truck Operator Costs:

- Operating cost: Labor, Fuel
- Travel time

Non-truck Operator Costs:

- Travel time

Benefit and cost categories not considered in this study but considered in a companion study:

Truck Operator Costs:

- Equipment Cost: Capital and maintenance (Approximated by the surrogate measure of travel time)

  NOTE: We assume that the differences in per-unit equipment cost among the three alternatives are insignificant. In particular, the difference between the cost of a conventional tractor and that of an AHS tractor is insignificant. Under this assumption, the cost of equipment can be approximated by a surrogate measure: truck equipment requirement or equivalently truck travel time.

  We consider the societal cost regardless of the organization against which the cost is incurred. For example, in the conventional alternatives, the entire truck equipment cost is incurred against the truck operator while in the truck AHS alternative, the truck-tractor portion of the equipment cost is first incurred directly against the truck-AHS operator and then passed onto to the freight forwarder. Although the truck-AHS operator may charge a service fee that is higher than the cost to make a profit, we assume that the profit reflects the efficiency gain achievable by the economy of scale associated with the concentration of investment in and operating expenses for long-haul automation and that the difference between the profit and the efficiency gain is insignificant.

Infrastructure Costs:

- Mainline
- Staging areas and other access and egress facilities

The deployment site serving as a reality check, not as a high-fidelity case study

Use I-5 from the California-Mexico border to the Oregon-California border for reality check.

In essence, the evaluation and comparison is decoupled into two separate and parallel but coordinated activities. The primary link between the two is the freight corridor. Particular corridor characteristics of importance include:

- Conventional freeway:
  - homogeneous freeway segments:
    - The default would be the segments corresponding to the truck-volume data (See Appendix A.1.)
  - corresponding numbers of lanes on each of the segments (See Appendix A.1.)
  - average traffic volumes: both truck volume and volume of other vehicles (See Appendix A.1.)
- Truck-AHS
  - The locations of AHS access and egress, in post miles from the border between California and Mexico (See Appendix A.2.)

THE METHODOLOGY: THE INPUT

Current corridor data

- For every homogeneous segment of Interstate 5 in California:
  - Length (See Appendix A.1.)
  - Number of lanes (See Appendix A.1)
  - Daily volumes on the segment: trucks vs. others (AM Peak, PM Peak, Off-peak vs. Night to be considered in the algorithm) (See Appendix A.1.)

Truck-AHS Design Data

- Locations of AHS interface in post miles (See Appendix A.2.)

Truck-Lane Design Data

- Locations of Truck-Lane interface in post miles (Same as AHS Interface locations; see Appendix A.2.)

Demand Inflation Factors: 100%, 125% and 150%. Current demand will be inflated by these percentages, and all the analyses performed for the current demand will be repeated for these percentages. (Since two of the three alternatives involve no additional capacity for non-truck and truck traffic accounts for a small fraction of the overall traffic, inflating the current demand to any higher level than 150% would give disproportionate advantages to the two alternatives in terms of efficiency gain in trucking but disproportionate advantages to the general-use lane alternative in terms of efficiency gain in overall travel time, including truck and non-truck travel time.) Also, the total truck demand after the inflation should not exceed the capacity of the one-lane truck-AHS or the one truck lane. This is a basic assumption of our analysis.

THE METHODOLOGY: ASSUMPTIONS

Assumptions:

- The local portions of the benefits and costs associated with the three alternatives are assumed to be equal, and they are cancelled in the comparisons. It suffices to focus on the mainline.
- The differences in per-unit equipment cost among the three alternatives are insignificant. In particular, the difference between the cost of a conventional tractor and that of an AHS tractor is insignificant. Under this assumption, the cost of equipment can be approximated by a surrogate measure: equipment requirement or equivalently truck travel time.
- We consider the societal cost regardless of the organization against which the cost is incurred.
- We assume that the non-truck traffic varies with respect to the hour of the day but that truck traffic does not. The truck arrivals at their on-ramps are assumed to be evenly distributed across the whole day in a deterministic fashion. An algorithm disaggregating the average daily non-truck demand among AM/PM Peak, Off-peak vs. Night has been developed. (This algorithm will be used for each of the three demand levels, i.e., 100%, 125% and 150% of the current demand.) Since the theoretical maximum for hourly flow rate is approximately 2100 vehicles per lane per hour, the number of peak hours may depend on the traffic demand. The algorithm will be discussed in detail later as part of **THE METHODOLOGY: A HIGH-LEVEL INTRODUCTION TO THE APPROACH**

Several design options selected are to be discussed later.

THE METHODOLOGY: ESTIMATION OF ORIGINS AND DESTINATIONS USING THE MAXIMUM-ENTROPY APPROACH

Scope: Origins and destinations refer to the start and end points of the trips on the mainline, i.e., the origin and destination ramps. They do not refer to the locations of the shippers' shipping docks and the consignees' receiving docks, respectively. Estimate origin and destination trip numbers based on the Interstate 5 truck volume data using approach of maximum-entropy approach.

The primary reason for estimating the OD trip numbers is to estimate the number of trucks that will use a truck-AHS, for the truck-AHS alternative, or a truck lane, for the truck-lane alternative. Therefore, we need to estimate the OD trip numbers of trucks only, and we do not need to estimate the OD trip numbers for the non-truck vehicles.

This methodology works in a variety of ways. The only data available to us are the number of trucks and the number of other vehicles traveling a section of Interstate 5, and no origin-destination trip numbers were available, and, therefore, we had to estimate them. For ease of discussion, we refer to these numbers simply as truck link counts and non-truck link counts.

We used the method of linearly-constrained constrained entropy maximization. In short, entropy can be interpreted as the amount of uncertainty contained in a distribution. The more uncertainty there is in a distribution, the higher the entropy. In a more visual term, the flatter the distribution or, equivalently, the wider spread, the higher the entropy. In this current context, a practically infinitely many possible origin-destination trip numbers can result in the observed truck link counts. The method of linearly-constrained maximum entropy produces the one (out of the practically infinitely many possibilities that result in the link counts) that is the most uncertain, is the flattest or, equivalently, has the widest spread.

This technique has been used for estimating the trip volumes associated with different pairs of origin-destination zones in a region (Fang and Tsao, 1995). In that case, the method is closely related to the so-called "gravity model." The method has also been used in studying the collision probability and impact force in the context of AHS safety by Tsao and Hall (1994). The methodology of entropy optimization has been treated recently in Fang, Tsao and Rajasekera (1997).

Linearly-constrained entropy maximization involves the generic entropy function as the objective function and a set of linear constraints. The only input required for a linearly-constrained entropy maximization problem is the linear constraints. A C program that produces the constraint set based the on and off volumes at all the stations has been developed.. A set of FORTRAN and C programs that solve the resulting linearly-constrained entropy maximization problem has also been developed.

THE METHODOLOGY: THE OUTPUT

Given a set of OD trip volume data along the corridor (mainline only), a set of computer program will be developed in C to provide, for each of the three options

- Addition of a truck-AHS lane
- Addition of a truck-only lane
- Addition of a general-use lane,

the following numerical values:

For trucks:

Operating cost category: Applicable for all three alternatives

- Travel time
  - Measure: total number of truck-revenue-hours - <u>one number</u>
- Operating cost:
  - Labor
    - Measure: the total number of driver-revenue-hours - <u>one number</u> (This is different from truck-revenue-hours for AHS because driverless operation is involved in and only in AHS.)
  - Fuel
    - Measure: the total amount of fuel - <u>one number</u>
- Equipment cost: Capital and Maintenance
  - Measure: the total number of truck-revenue-hours - same as travel time measure

(Note: The travel time and labor cost may be

- incurred on AHS, and/or
- incurred on regular lanes or the truck lane

If this difference is significant, then we will need to refine the travel time and labor costs into four more refined components:

- AHS Travel Time
- Conventional Freeway Travel Time
- AHS labor
- Conventional Trucking Labor.)

For non-trucks:

- Travel time
  - Measure: total number of non-truck-hours - <u>one number</u>

Four Numbers as the Output: Five Performance Values (Truck Time and Equipment Requirement are assumed equivalent)

<u>For Non-trucks</u>:

- Travel Time: Total Non-Truck Travel Time: non-truck-hour

<u>For Truck</u>:

- Travel Time and Equipment Requirement - Total Truck Travel Time: truck-revenue-hour
- Labor - Total Truck Labor = Total Truck Travel Time: driver-revenue-hour
- Fuel - Total Truck Fuel: liter

Note: Equipment requirement can also be measured in terms of total distance traveled by trucks, in the unit of truck-revenue-miles. Although this measure can be calculated, this total distance does not depend on the alternatives, and does not distinguish one alternative from another.

THE METHODOLOGY: A HIGH-LEVEL INTRODUCTION TO THE APPROACH

The overall performance measures of the truck-AHS results from two related sets of measures estimating the performance of the system on two different components: (a) the conventional lanes and (b) the truck or truck-AHS lanes. Given the knowledge of how the overall demand is split between the two components, the two sets of performance measures can be estimated. A key question is how to split the demand between the two components. Our approach is to first estimate the amount of truck traffic that will be attracted away from the conventional lanes onto the truck or truck-AHS lanes, and given the split traffic we solve two independent problems, one dealing with the conventional lanes and the other dealing with the truck or truck-AHS lane. As mentioned earlier, we assume that the arrival rate of non-trucks at their corresponding on-ramps varies with respect to the hour of the day but that its truck counterpart does not.

Since the conventional alternative of building a general-use lane involves only conventional lanes and both of the other alternatives also involve conventional lanes, we first discuss how to estimate the performance of a conventional freeway, given the traffic demand for the conventional lanes. We will then discuss the estimation of performance measures related to the truck or truck-AHS lane.

We consider only three levels of service and the three corresponding periods: AM/PM Peak, Near Peak and Free Flow. AM/PM Peak is assumed to be characterized by a flow rate of 2100 automobile-equivalents per hour at the speed of 35 miles per hour; Near Peak is assumed to be characterized by a flow rate of 1800 automobile-equivalents per hour at the speed of 50 miles per hour; Free Flow is assumed to be characterized by a flow rate of 1500 or below with a speed of 75 miles per hour. We first estimate the number of AM/PM Peak hours, and assume that the number of Near Peak hours is the maximum of 0 and half of the number of AM/PM Peak hours. The rest of the 24 hours, if any, are Free-Flow hours. For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or a Near Peak hour is obtained by subtracting the average number of truck traveling the section per hour times 2 (to obtain the automobile-equivalents) from the flow rate of the corresponding hour. Based on these numbers, the three assumed speeds and the section lengths, both the total per-day truck travel time and total per-day non-truck travel time can be calculated.

This is how we estimate the number of AM/PM Peak hours in a day. Given daily section traffic counts, including the truck counts and the total counts, we first estimate the average number of automobile-equivalents per lane per hour (for one direction only but averaged over 24 hours) for each freeway section. Only two vehicle types are considered: trucks and non-trucks; one truck is considered as two automobile equivalents. We assume that if and only if, for any given section, the number of automobile-equivalents per lane per hour exceeds 500, then the section experiences peak hour congestion. For each additional 67 automobile-equivalents per lane per hour, the number of peak hours is increased by 1. If the number of automobile-equivalents per lane per hour is 2100 averaged over a 24-hour day, then this calculation produces 24 peak hours for the corresponding section. This is reasonable because in such a case, the demand of the section is so high that it is saturated all day. Such saturation occurs when the current demand is inflated to study the performance of the three alternatives in the future.

Truck traffic is assumed to be evenly distributed across the 24 hours of a day in a deterministic fashion. For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or Near Peak hour is obtained by subtracting the average hourly number of truck traveling the section times 2 (to obtain the automobile-equivalents) from the flow rate of the corresponding hour. The number of non-trucks that travel the section at the free flow can be obtained by subtracting the corresponding numbers for the Peak and Near Peak hours from the total daily non-truck count. Based on these numbers, the three assumed speeds and the section lengths, both the total per-day truck travel time and total per-day non-truck travel time can be calculated.

Calculations for the fuel consumption and labor requirement for operations on the conventional lanes are straightforward. The labor requirement is exactly the same as the truck travel time. The fuel consumption is estimated based on the following average gas-mileages. 10 miles per gallon for trucks, also regardless of speed.

We now address the performance associated with the other component, i.e., the truck-AHS or the truck lane.

The problem of determining number of trucks that will be attracted away from the conventional lanes onto the truck-AHS or the truck lane is too big to be solved as one modeling-optimization problem. We use a simpler approach with the following characteristics. For each of the two alternatives (i.e., the truck lane or truck-AHS), our approach is similar. For the Truck-AHS option (or the truck-lane option), a truck operator is offered two choices: use AHS (or the truck lane) or not use AHS (or the truck lane). We summarize the approach using the context of truck-AHS. The approach is characterized by:

- Parameterized Decision Rules: The decision to be made by a trucking operator as to whether to use AHS or not is a very complex one, depending on the cost and travel-time advantages. We consider several decision rules, and each of the rules is parameterized at discrete levels.

- Estimate the recurring performance values resulting from the satisfaction of the OD demand for <u>one day</u>. (To get the recurring performance values for one year, for example, just multiply these values by 365.)

- Selection one or more most appropriate sets of the parameters as the optimal operational designs for truck-AHS and as the base for truck-AHS benefit-cost calculation. This can be enhanced in the future to include other important considerations, e.g., infrastructure costs and capital and maintenance costs of vehicles.

THE METHODOLOGY: TRUCK-AHS DESIGN OPTIONS, OPERATING RULES AND EVALUATION ASSUMPTIONS SELECTED

Design Option: Closed-system Operating Concept

There exist many design options for a truck-AHS. This methodology has been developed for a closed-system operating concept only. A study of an open-system operating concept involves the technology decision on the part of a trucking operator whether to equip its trucks for AHS operations. Making such decisions requires careful consideration of many other factors that are out of the scope of this project. In a closed-system operating concept, trailers are practically the primary units of freight movement and they can be hauled by either regular tractors off-AHS or by AHS tractors on AHS. To move trailers onto AHS, detach the trailers from the regular tractors and attach them to AHS tractors. For ease of discussion, we refer to trucks as the primary units of freight movement on a truck-AHS, although trailers actually are the primary units and the regular tractor will be substituted with an AHS tractor when traveling on the truck-AHS.

Design Option: Dynamic Convoying, with Infrastructure-supervised Merging and Splitting at or near AHS Access/Egress Points

A convoy may consist of trucks destined for multiple AHS exits. Trucks may merge into or split off safely from the convoy but only at or near AHS access/egress points, where roadway conditions are closely monitored by the infrastructure. In this way, only those trucks destined for an AHS egress point will need to leave the mainline and enter the corresponding staging area.

Rule 1: End-to-end Shuttle Convoy; Fixed Headway: Run an end-to-end shuttle truck with a driver aboard between the two end points with a fixed headway. In our case, run a shuttle between the California-Mexico border and the California-Oregon border. The headway is selected so that the maximum convoy size, i.e., the maximum number of trucks in a convoy, is not excessive. In our study, 20 is considered a target maximum size, but in some busy AHS sections, the convoy size is actually larger than 20.

Rule 2: All trucks traveling in a Convoy, with a Driver onboard only the Lead Truck: All trucks traveling on the truck-AHS must be attached to a closely spaced convoy, and every such convoy must be led by an end-to-end shuttle truck. Truck-following is automated and is driverless.

Rule 3: Maximum Convoy Size, with Flexibility for Larger Size to Cope with Demand Volatility. For safety reasons, set a maximum convoy size. As just mentioned, we use 20 as the target maximum size. However, to cope with possibly volatile demand, actual size of a convoy may be allowed to exceed 20 depending on circumstances.

Rule 4: An Entering Truck or Convoy Joins the Next Passing Shuttle Convoy. Trucks arriving at a truck-AHS access point will join the next passing shuttle convoy on the mainline after a proper mode change. If there are more than one truck that wish to enter the AHS, organize these trucks into a convoy and the convoy will enter the AHS by joining the next passing truck convoy already traveling on the AHS.

Evaluation Assumption: AHS Usage Decision Rule - Trip Length on Mainline

AHS USAGE THRESHOLD: If the mainline portion of the trip of a truck exceeds PARAMETER - AHS USAGE THRESHOLD, it will use the AHS. Otherwise, it will not.

Evaluation Assumption: Circuitous Back Travel Rule - Relative Closeness to the previous and the Next AHS Access Points

Sometime, it is worthwhile for a truck to go backward away from the destination so that it can access the truck-AHS much more quickly than going forward toward the destination (to enter the truck-AHS via the next access point). Similarly, sometimes it may be worthwhile for truck traveling on the AHS to go past the destination (conventional) freeway exit, leave the truck-AHS at the next AHS egress point, and then travel backward toward the destination (conventional) freeway exit. The following parameter is used to decide whether a truck should go backward or not.

BACK TRAVEL THRESHOLD: If a truck will use the AHS (according to the previous rule) and the distance from the entry point of the truck on the (conventional) freeway to the previous AHS access point divided by the total distance between the previous and the next AHS access points is less than BACK TRAVEL THRESHOLD, then the truck will travel backward to the previous AHS access point so as to access the AHS sooner. Otherwise, it will travel to the next AHS access point and then enter the truck-AHS there.

Similarly, if the distance between the destination (conventional) freeway exit of a truck and the first AHS egress point beyond that destination divided by the distance between that AHS egress point and the AHS egress point immediately before it is less than BACK TRAVEL THRESHOLD, the truck will travel to the AHS egress point beyond the destination (conventional) freeway exit first before traveling in reverse direction toward the destination (conventional) freeway exit.

THE METHODOLOGY: A MORE DETAILED DESCRIPTION

The regular-lane alternative involves only travel on conventional lanes, but both of the other two alternatives involve both conventional (general-use) lanes and a new type of lane. We use a group of algorithms to estimate the performance values associated with the traffic traveling on the conventional lanes and another group of algorithms to estimate the performance values associated with the traffic using an AHS lane. We also develop a third group of algorithms to estimate the performance values associated with using a truck-lane. However, the third group is very similar to the second. We will discuss only the differences. (There are big differences in actual operations though, and big performance differences are expected.)

For the truck-AHS and the truck-lane options, the total performance values are merely the sum of the performance values obtained for the portion of the conventional lanes and those obtained for the new lane type. For the conventional-lane option, the total performance value is simply the performance value obtained using the algorithms developed for the conventional lanes itself (<u>but with one less traffic lane</u>). We now discuss the three groups of algorithms.

Algorithms Estimating the Performance Values of the General-use-lane Alternative

For the conventional-lane option, there is no need to split the total OD demand between the conventional freeway and any other lane type. But, for the other two alternatives, splitting the total demand into two is required. The way to split depends on the lane type, and will be addressed in the next two groups of algorithms.

For the rest of the discussion about this algorithm group, we assume that the total daily demand for every section of the conventional freeway is given. In addition, the daily demand is split into the truck volume and the non-truck volume.

Estimation of Travel Time

Step 1: Estimating the number of AM/PM Peak hours, the number of Near Peak hours and the number of Free Flow hours.

Only three levels of service and the three corresponding periods are considered: AM/PM Peak, Near Peak and Free Flow. AM/PM Peak is assumed to be characterized by a flow rate of 2100 automobile-equivalents per hour at the speed of 35 miles per hour; Near Peak is assumed to be characterized by a flow rate of 1800 automobile-equivalents per hour at the speed of 50 miles per hour; Free Flow is assumed to be characterized by a flow rate of 1500 or below with a speed of 75 miles per hour. We first estimate the number of AM/PM Peak hours, and assume that the number of Near Peak hours is the maximum of 0 and half of the number of AM/PM Peak hours. The rest of the 24 hours, if any, are Free-Flow hours. For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or a Near Peak hour is obtained by subtracting the average number of truck traveling the section per hour times 2 (to obtain the automobile-equivalents) from the flow rate of the corresponding hour. Based on these numbers, the three assumed speeds and the section lengths, both the total per-day truck travel time and total per-day non-truck travel time can be calculated.

The number of AM/PM Peak hours in a day is estimated as follows. Given daily section traffic counts, including the truck counts and the total counts, we first estimate the average number of automobile-equivalents per lane per hour (for one direction only but averaged over 24 hours) for each freeway section. Only two vehicle types are considered: trucks and non-trucks; one truck is considered as two automobile equivalents. We assume that if and only if, for any given section, the number of automobile-equivalents per lane per hour exceeds 500, then the section experiences peak hour congestion. For each additional 67 automobile-equivalents per lane per hour, the number of peak hours is increased by 1. If the number of automobile-equivalents per lane per hour is 2100 averaged over a 24-hour day, then this calculation produces 24 peak hours for the corresponding section. This is reasonable because in such a case, the demand of the section is so high that it is saturated all day. Such saturation occurs when the current demand is inflated to study the performance of the three alternatives in the future.

Step 2: Estimate the number of trucks using each section during AM/PM Peak, Near Peak and Free Flow hours.

Truck traffic is assumed to be evenly distributed across the 24 hours of a day in a deterministic fashion.

Step 3: Estimate the number of non-trucks using each section during AM/PM Peak, Near Peak and Free Flow hours.

For any section of the freeway, the number of non-trucks (i.e., automobiles) that can be accommodated within a AM/PM Peak hour or a Near Peak hour is obtained by subtracting the average number of trucks traveling the section per hour times 2 (to obtain the automobile-equivalents) from the total flow rate of the corresponding hour. The total number of non-trucks that travel the section at the free flow can be obtained by subtracting the corresponding total numbers for the Peak and Near Peak hours from the total daily non-truck count.

Step 4: Obtain the per-day Total Truck Travel Time in truck-revenue-hour and the per-day Total Non-truck Travel Time in non-truck-hour.

The section length divided by the three different speeds produces the travel time of the section during the three different periods. The section travel time multiplied by the total the number of vehicles (for either trucks or non-trucks) traveling on it during any of the three periods produces the total Travel Time spent during the three corresponding periods. The sum of the three numbers is the Total Travel Time (for either trucks or non-trucks).

Estimation of Labor Requirements

Step 5: The value of Total Truck Labor (in driver-revenue-hour) is the same as the Total Truck Travel Time (in truck-revenue-hour). (This is because one driver is required for every truck.)

Estimation of Truck Fuel Requirements

Step 6: For each of the 24 one-hour intervals and each of the segments, determine the per-unit-distance fuel consumption at the corresponding speed from a table, and multiply it by the volume for the hour and by the length of the segment. For simplicity, the fuel consumption is estimated based on the following average gas-mileages: 10 miles per gallon for trucks, also regardless of speed.

Step 7: Obtain the sum, and the sum is the Total Truck Fuel (Requirements) (in liter).

These algorithms will produce the following performance values for this option:

For Non-trucks:

- Travel Time: Total Non-truck Travel Time: non-truck-hour – one number

For Truck:

- Travel Time: Total Truck Travel Time: truck-revenue-hour – one number
- Labor: Total Truck Labor: driver-revenue-hour = Total Truck Travel Time – one number
- Fuel: Total Truck Fuel (Requirement): liter – one number

as well as an overall weighted sum – one number.

The Final Output Template for the General-use-lane Alternative:

	Current Demand	125% Inflation	150% Inflation
Total Non-truck Travel Time: non-truck-hour			
Total Truck Travel Time: truck-revenue-hour			
Total Truck Labor: driver-revenue-hour			
Total Truck Fuel (Requirement): liter			

Algorithms Estimating the Performance Values of the Truck-AHS Alternative

For the alternative of truck-AHS, the performance values will be the sum of the corresponding values for the conventional lanes and those for the truck-AHS lane.

The general approach is to estimate all four performance measures as a function of the parameters defining the two key parameterized decision rules. A set of the most appropriate parameter values will be chosen and the resulting performance values estimated.

Step 0: For every selected parameter set, perform the following steps:

The decision rules involve the following parameters:

PARAMETER - AHS USAGE THRESHOLD
PARAMETER - BACK TRAVEL THRESHOLD
PARAMETER - AHS HEADWAY

We now describe how to obtain the four performance values based on a given set of parameter values. We first deal with the AHS portion, and then the portion of conventional freeway. This is because the decision-rules for AHS usage determines how much of the overall truck traffic will be diverted to the truck-AHS.

Truck-AHS Portion

Step 1: Determine the OD pairs for which the trucks will use the truck-AHS, using PARAMETER - AHS USAGE THRESHOLD.

Given the daily truck trip number between two specific mainline freeway access/egress points, use two decision rules discussed earlier (as design options) to determine if all the corresponding trucks will use the AHS, and if so, whether they will go against the direction of the destination so as to use the AHS as much as possible.

If the mainline portion of the trip of a truck exceeds PARAMETER - AHS USAGE THRESHOLD, it will use the AHS. Otherwise, it will not.

Step 2: If so, determine if truck will travel backward or beyond to fully utilize the truck-AHS, using PARAMETER - BACK TRAVEL THRESHOLD.

If a truck will use the AHS (according to the previous rule) and the distance from the entry point of the truck on the (conventional) freeway to the previous AHS access point divided by the total distance between the previous and the next AHS access points is less than BACK TRAVEL THRESHOLD, then the truck will travel backward to the previous AHS access point so as to access the AHS sooner. Otherwise, it will travel to the next AHS access point and then enter the truck-AHS there.

Similarly, if the distance between the destination (conventional) freeway exit of a truck and the first AHS egress point beyond that destination divided by the distance between that AHS egress point and the AHS egress point immediately before it is less than BACK TRAVEL THRESHOLD, the truck will travel to the AHS egress point beyond the destination (conventional) freeway exit first before traveling in reverse direction toward the destination (conventional) freeway exit.

Step 3: Given the (daily) OD trip numbers for the truck-AHS calculated in the previous two steps for all OD pairs, determine the number of trucks attracted away from a conventional freeway section in a day, and the remaining counts will be the traffic to travel on the conventional freeway. Also calculate the (daily) truck counts for all the segments on the AHS, which is different from the number of trucks attracted away from a conventional freeway section in a day

Conventional Freeway Portion:

Step 4: Determine the performance values associated with the freeway portion of the traffic (for one day) according to the algorithms stated earlier for the conventional-freeway option. Note, however, this time the number of lanes is exactly the current number of lanes (without the additional general-use lane).

The performance values include:

For Non-trucks:
- Total Non-truck Travel Time: non-truck-hour

For Truck:
- Total Truck Travel Time: truck-revenue-hour
- Total Truck Labor: driver-revenue-hour
- Total Truck Fuel (Requirement): liter

Back to the AHS Portion: Estimation of Travel Time on AHS, Truck Labor, Truck Fuel and Staging Area.

Step 5: Determine the total travel time spent on the AHS by multiplying the travel time associated with a particular AHS OD and the corresponding OD counts and add a constant expected delay at the staging area that is equal to one half the PARAMETER - AHS TRACTOR

HEADWAY plus AHS MODE CHANGE TIME. In addition, add the travel time of the end-to-end shuttle trucks.

Note that it is assumed that the AHS traffic moves at the design speed because the capacity of the AHS is (assumed to be) higher than the demand. PARAMETER - AHS MODE CHANGE TIME is assumed to be constant; its value is set to be 15 minutes in our numerical study. This can be improved. The time required for the mode change may depend on the volume. But, if the capacity of the mode change facility can be adjusted so that the average mode change time remains approximately constant, this assumption is reasonable. However, the cost of providing such flexibility must also be counted (somewhere else in the estimation process).

A constant headway is assumed, but this assumption can be relaxed in the future as an improvement to the current algorithms. Also, currently, the AHS shuttle tractors travel from one end of the corridor to the other. Another improvement could be to specify a number of "partial routes" involving only a portion of the corridor and the companion headways.

Step 6: Determine the total per-day truck labor required to operate the truck-AHS.

Note the following when calculating the performance value. At the beginning of a day, AHS tractors will be at different locations along the corridor, and, at the end of a day, these or other AHS tractors will also be at different locations along the corridor. Also note that only one driver is required per convoy. (The size of convoy is currently not controlled. The convoy size distribution will be calculated; some of the parameter set may be considered unacceptable if the resulting convoy size is too large. The convoy size distribution is to be addressed below.)

The following parameter plays a key role in this step.

PARAMETER - AHS SHUTTLE TRACTOR HEADWAY

Note that this requirement is calculated for one day of operations; the requirement for travel time is calculated for satisfying one-day worth of demand. Although the bases are different, they both represent per-day costs.

Step 7: Obtain the convoy size for each section, and calculate the fuel consumption.

Because of the constant headway, the convoy size for each of the convoys within each of the segments can be easily calculated. It is simply the size of the convoy within the previous segment minus the number of trucks leaving the AHS plus the number of trucks entering the AHS. Note that each convoy is led by an end-to-end shuttle truck, which may not haul any freight. For our evaluation purposes, we assume that such shuttle trucks do not haul any freight.

Note that we need to keep track of how much time a convoy remains in a particular size. Therefore, the overall convoy size distribution must also be a weighted average with respect to the time spent in the different sizes.

We assume that trucks arriving at a particular (conventional) freeway access point and destined for a particular (conventional) freeway egress point are equi-spaced in time (in a deterministic fashion) but depend on the OD volume, of course. (This can be improved later. The improvement should not be difficult. We need to first generate the actual but random arrival times. The program needs to be modular enough so that this improvement can be implemented with ease.) A convoy travels through the corridor, and picks up all the waiting trucks.

These algorithms will produce the following performance values for the truck-AHS alternative.

For Non-trucks:

- Total Non-Truck Travel Time: non-truck-hour – one number

For Truck:

- Total Truck Travel Time: truck-revenue-hour – one number
- Total Truck Labor: driver-revenue-hour – one number
- Total Truck Fuel: liter – one number

as well as an overall weighted sum of the first four (recurring performance values) – one number.

<u>The Output Template for the Truck-AHS Alternative:</u>

	Current Demand	125% Inflation	150% Inflation
Total Non-truck Travel Time: non-truck-hour			
Total Truck Travel Time: truck-revenue-hour			
Total Truck Labor: driver-revenue-hour			
Total Truck Fuel (Requirement): liter			

Templates for Key Intermediate Results:

Truck AHS OD Intermediate Results

Truck OD Trip Numbers: (Daily Volume; One Direction Only)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck OD Trip Numbers Using the Truck AHS: (Daily Volume; One Direction Only; 0 Representing Non-use)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck ODs Not Using the Truck AHS: (Daily Volume; One Direction Only)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck OD Trip Numbers on Truck-AHS: (Hourly; One Direction Only)

Truck AHS Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck AHS Link Volume (Link = Segment = Section)

Truck Volumes on Truck-AHS Sections (One Direction Only; Daily; Aggregated Over Daily OD Trip Numbers) and

AHS Section	Truck Volume

Conventional Freeway Link Volume (with Truck AHS Link Volume Subtracted)

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Daily Total, Obtained by Subtracting the Truck AHS Link Volume Values from the Original and Raw Data on Overall Traffic LINK Demand (One Direction Only; Daily)

Freeway Section	Truck Volume	Non-Truck Volume	Total Volume

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – AM/PM Peak (One Direction Only; Hourly)

Freeway Section	Truck Vol. - P	Non-Truck Vol. - P	Total Volume - P

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Near Peak (One Direction Only; Hourly)

Freeway Section	Truck Vol. - NP	Non-Truck Vol. - NP	Total Volume - NP

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Free Flow (One Direction Only; Hourly)

Freeway Section	Truck Vol. - FF	Non-Truck Vol. - FF	Total Volume - FF

For every parameter set:

For One AM/PM Peak Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For One Near Peak Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For One Free Flow Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For truck-AHS:

Basic Truck AHS Link Information (One Direction; Daily)

AHS Section #		
Length of Section		
Volume of Section – Daily: # Trucks (Derivable from AHS OD Trip Numbers)		
Speed of Section – Design Speed: Miles per Hour		

Basic AHS Access/Egress Points

AHS Interface #		
Location		

Truck AHS Routes and Schedules: Only one route is considered – Shuttle from one end to the other; only one uniform headway is considered.

Truck AHS Routes and Schedule (One Direction)

Route #		
Route Origin		
Route Destination		
Headway		

Note: Currently, we accommodate only end-to-end shuttle trucks, with a fixed headway. In the future, this can be extended to incorporate "partial routes." Keep route structure and schedule separate from other parts of the program, at least as much as possible, because new routes and schedules may be studied as we gain experience on the relationship of performance of the AHS as a function of the routes and schedules. Also, keep the route structure and the schedule separate from each other if possible.

Algorithms Estimating the Performance Values of the Truck-lane Alternative

Operate the system the same way as the truck-AHS to the maximum possible extent. However, convoying and driverless operations cannot be done.

All the steps are identical to those of the algorithms for the truck-AHS alternative, except for the following:

Step 6: Determine the total per-day truck labor required to operate the truck-lane.

This step is much simpler than its AHS counterpart because one driver is required for each truck using the truck lane. (No end-to-end shuttle truck is involved either.)

Step 7 is not needed because this alternative involves no convoying.

All the other steps are the same except that the AHS lane is replaced with the truck lane. Also note that the parameter values may be different.

These algorithms will produce the following performance values for the truck-lane alternative.

For Non-trucks:

- Total Non-Truck Travel Time: non-truck-hour – one number

For Truck:

- Total Truck Travel Time: truck-revenue-hour – one number
- Total Truck Labor: driver-revenue-hour – one number
- Total Truck Fuel: liter – one number

as well as an overall weighted sum – one number.

<u>The Output Template for the Truck-lane Alternative:</u>

	Current Demand	125% Inflation	150% Inflation
Total Non-truck Travel Time: non-truck-hour			
Total Truck Travel Time: truck-revenue-hour			
Total Truck Labor: driver-revenue-hour			
Total Truck Fuel (Requirement): liter			

Templates for Key Intermediate Results:

Truck Lane OD Intermediate Results

Truck OD Trip Numbers: (Daily Volume; One Direction Only)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck OD Trip Numbers Using the Truck Lane: (Daily Volume; One Direction Only; 0 Representing Non-use)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck ODs Not Using the Truck Lane: (Daily Volume; One Direction Only)

Freeway Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck OD Trip Numbers on Truck Lane: (Hourly; One Direction Only)

Truck Lane Origin\Destination	Access Point 1	Access Point 2	…….	Access Point n
Access Point 1				
Access Point 2				
……				
Access Point n				

Truck Lane Link Volume (Link = Segment = Section)

Truck Volumes on Truck-AHS Sections (One Direction Only; Daily; Aggregated Over Daily OD Trip Numbers) and

Truck Lane Section	Truck Volume

Conventional Freeway Link Volume (with Truck Lane Link Volume Subtracted)

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Peak and Non-Peak Daily Total, Obtained by Subtracting the Truck Lane Link Volume Values from the Original and Raw Data on Overall Traffic LINK Demand (One Direction Only; Daily)

Freeway Section	Truck Volume	Non-Truck Volume	Total Volume

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Peak (One Direction Only; Hourly)

Freeway Section	Truck Vol. - P	Non-Truck Vol. - P	Total Volume - P

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Near Peak (One Direction Only; Hourly)

Freeway Section	Truck Vol. - NP	Non-Truck Vol. - NP	Total Volume - NP

Truck Volume and Non-truck Volume Using the Conventional Freeway Lanes – Free Flow (One Direction Only; Hourly)

Freeway Section	Truck Vol. - FF	Non-Truck Vol. - FF	Total Volume - FF

For every parameter set:

For One AM/PM Peak Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For One Near Peak Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For One Free Flow Hour on the Conventional Freeway: (One Direction; One Hour)

Section #		
Length of Section		
Volume: # Trucks		
Volume: # Other Vehicles		
Volume: # All Vehicles		
Speed of Section		
Travel Time - per Vehicle (= Section Length/Speed)		
*Total Non-Truck Travel Time: non-truck-hour		
*Total Truck Travel Time: truck-revenue-hour		
*Total Truck Labor = Total Truck Travel Time: driver-revenue-hour		
*Total Truck Fuel: liter		

For Truck Lane:

Basic Truck Lane Link Information (One Direction; Daily)

Truck Lane Section #		
Length of Section		
Volume of Section – Daily: # Trucks (Derivable from Truck Lane OD Trip Numbers)		
Speed of Section – Design Speed: Miles per Hour		

Basic Truck Lane Access/Egress Points

Truck Lane Interface #		
Location		

APPENDIX C: SOFTWARE TOOLS FOR EVALUATION AND COMPARISON

APPENDIX C.1: C Program for the Max-Entropy Problem Generator

```
#include <stdio.h>

/* Some basic type declarations */
#define TRUE            1
#define FALSE           0

#define MAX_STATIONS 250


/* Default Input File names */
#define DISTANCE_FILE           "distance.txt"

#define SAMPLE_IN_SECTION_FILE      "section.txt"

/* Output file names */
#define DISTANCE_MATRIX         "distance.out"
#define SAMPLE_LHS_OUT_FILE     "data.a"
#define SAMPLE_RHS_OUT_FILE     "data.b"
#define SAMPLE_SIZE_OUT_FILE    "data.sizes"

/* Actual input file names */
char dist_file[80];

char in_section_file[80];

char out_lhs_file[80];
char out_rhs_file[80];
char out_size_file[80];

/* Forward declarations of functions */
void getInputFileNames(void);
void readDistFile(char filename[], float *dist_array, int stations);
void readDataFile(char filename[], int *array, int stations);
void generateDistMatrix(float *dist_matrix, float *dist_array, int stations);

void writeLhsConstraintFile(char lhs_file[], int stations);
void writeRhsConstraintFile(char rhs_file[], int *array, int stations);
void writeDataSizes(char size_file[], int stations);


main(int argc, char **argv)
{
    int i;
    int num_stations;
    float avg_trip_len, min_trip_len, max_trip_len;

    char buffer[100];

    /* Input Data Arrays */
    float dist_array[MAX_STATIONS];

    int section_data_array[MAX_STATIONS];
```

```
    /* Generated Matrix Data Structures */
    float dist_matrix[MAX_STATIONS][MAX_STATIONS];

    printf("\n");
    if (argc == 1) {
        do {
            printf("Enter the number of Stations: ");
            gets(buffer);
            sscanf(buffer, "%d", &num_stations);
        } while (strcmp(buffer, "") == 0 || num_stations <= 0);
    }
    else {
        printf("\nUSAGE: seperate_matrix\n\n");
        exit(0);
    }

    /* No need to run this program unless there are atleast 2 stations */
    if (num_stations < 2) {
        printf("Number of stations SHOULD be ATLEAST 2...\n\n");
        exit(0);
    }

    getInputFileNames();

    /* Read the Distance vector file */
    readDistFile(dist_file, dist_array, num_stations);

    readDataFile(in_section_file, section_data_array, num_stations);

    /* *** Generate LHS *** */

    /* Write LHS Section Constraints to the output file */
    writeLhsConstraintFile(out_lhs_file, num_stations);


    /* *** Write RHS *** */
    /* Write RHS Section data */
    writeRhsConstraintFile(out_rhs_file, section_data_array, num_stations);

    /* Write the data size */
    writeDataSizes(out_size_file, num_stations);
    printf("INFO: Wrote M,N data sizes to output file: %s\n", out_size_file);
    printf("\n");

}

void getInputFileNames(void)
{
    char buffer[100];

    /* Get the Distance vector file name from the user */
    printf("Enter the file_name for Distance between Stations [%s]: ",
                                                  DISTANCE_FILE);
    gets(buffer);
    sscanf(buffer, "%s", dist_file);
    if (strcmp(dist_file, "") == 0) {
        memcpy(dist_file, DISTANCE_FILE, strlen(DISTANCE_FILE));
    }
```

```
    /* Get the Section Data file name from the user */
    printf("Enter the input file name for Section Data [%s]: ",
                                      SAMPLE_IN_SECTION_FILE);
    gets(buffer);
    sscanf(buffer, "%s", in_section_file);
    if (strcmp(in_section_file, "") == 0) {
        memcpy(in_section_file, SAMPLE_IN_SECTION_FILE,
strlen(SAMPLE_IN_SECTION_FILE));
    }

    printf("Enter the output file name for LHS [%s]: ", SAMPLE_LHS_OUT_FILE);
    gets(buffer);
    sscanf(buffer, "%s", out_lhs_file);
    if (strcmp(out_lhs_file, "") == 0) {
        memcpy(out_lhs_file, SAMPLE_LHS_OUT_FILE,
strlen(SAMPLE_LHS_OUT_FILE));
    }

    printf("Enter the output file name for RHS [%s]: ", SAMPLE_RHS_OUT_FILE);
    gets(buffer);
    sscanf(buffer, "%s", out_rhs_file);
    if (strcmp(out_rhs_file, "") == 0) {
        memcpy(out_rhs_file, SAMPLE_RHS_OUT_FILE,
strlen(SAMPLE_RHS_OUT_FILE));
    }

    printf("Enter the output file name for sizes [%s]:
",SAMPLE_SIZE_OUT_FILE);
    gets(buffer);
    sscanf(buffer, "%s", out_size_file);
    if (strcmp(out_size_file, "") == 0) {

memcpy(out_size_file,SAMPLE_SIZE_OUT_FILE,strlen(SAMPLE_SIZE_OUT_FILE));
    }

    printf("\n");
}


void readDistFile(char filename[], float *array, int stations)
{
    int i;
    FILE *fp;

    if ((fp = fopen(filename, "r")) == NULL) {
        printf("\nERROR: Couldn't open Input file: %s\n\n", filename);
        exit(0);
    }

    /* Distance from station to itself is zero */
    array[0] = 0.0;

    i = 1;
    while (i < stations && fscanf(fp, "%f\n", &array[i]) != EOF) {
        i++;
    }
    fclose(fp);

    if (i != stations) {
```

```
        printf("\nERROR: Missing Data. Tried to read %d entries, "
               "but read only %d entries.\n\n", stations, i);
        exit(0);
    }
}

void readDataFile(char filename[], int *array, int stations)
{
    int i;
    FILE *fp;

    if ((fp = fopen(filename, "r")) == NULL) {
        printf("\nERROR: Couldn't open Input file: %s\n\n", filename);
        exit(0);
    }

    array[0] = 0.0;
    i = 1;

    while (i < stations && fscanf(fp, "%d\n", &array[i]) != EOF) {
        i++;
    }
    fclose(fp);

    if (i != stations) {
        printf("\nERROR: Missing Data. Tried to read %d entries, "
               "but read only %d entries.\n\n", stations, i);
        exit(0);
    }

}


void writeLhsConstraintFile(char lhs_file[], int stations)
{
    FILE *fp_lhs;
    int col, row, line;
    char mode[10];

    {
        strcpy(mode, "w");
    }

    if ((fp_lhs = fopen(lhs_file, mode)) == NULL) {
        printf("\nERROR: Couldn't open Output LHS file: %s\n\n", lhs_file);
        exit(0);
    }

/*Here, "line" means "section" while "row" and "col" refer to stations.  All
indices start from 0.*/

 for (line=0; line<stations-1; line++) {

        /* Generate the LHS of the equation */
        for (row=0; row<stations-1; row++) {
            for (col=row+1; col<stations; col++) {

                {
                    if (line >= row && col >= line+1) {
```

```
                        fprintf(fp_lhs, "%d ", 1);
                    }
                    else {
                        fprintf(fp_lhs, "%d ", 0);
                    }
                }

            }
        }
        /*fprintf(fp_lhs, "\n");*/
    }

    fclose(fp_lhs);
}


/*NOTE:  Make sure that the section demand data are stored as 0, section 1
demand, section 2 demand, …, last section demand.  This way, the number of
entries is the number of stations, yet the number of sections is one less and
the section demand data begin at index = 1, i.e., the 2nd entry in the
array.*/

void writeRhsConstraintFile(char rhs_file[], int *array,
                            int stations)
{
    FILE *fp_rhs;
    int line;
    char mode[10];
    int start, end;

    {
        strcpy(mode, "w");
        start = 1;
        end   = stations;
    }


    if ((fp_rhs = fopen(rhs_file, mode)) == NULL) {
        printf("\nERROR: Couldn't open Output RHS file: %s\n\n", rhs_file);
        exit(0);
    }

    for (line=start; line<end; line++) {
        /* Generate the RHS of the equation */
        /*fprintf(fp_rhs, "%d\n", array[line]);*/
        fprintf(fp_rhs, "%d ", array[line]);
    }

    fclose(fp_rhs);
}

void writeDataSizes(char size_file[], int stations)
{
    FILE *fp;
    int constraints = 0, variables = 0;
    constraints = stations - 1;
    variables   = stations * (stations -1) / 2;

    if ((fp = fopen(size_file, "w")) == NULL) {
```

```
        printf("Couldn't write to %s\n", size_file);
        exit(0);
    }
    /*fprintf(fp, "%d,%d\n", constraints, variables);*/
    fprintf(fp, "%d %d", constraints, variables);
    fclose(fp);
}
```

APPENDIX C.2: PERFORMANCE ESTIMATION FOR THE AHS-LANE PORTION OF THE AHS ALTERNATIVE

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define MAXLEN 200       /* max size of number */
#define AHS_DIST  150              /* distance for a truck to take AHS lane
*/
#define BACKTHRESHOLD 0.1       /* threshold for a truck to go backwards for
an AHS exit */

double distance[MAXLEN];
int station_num[MAXLEN],AHSexitnum[16], truck_length[MAXLEN],
traffic_length[MAXLEN], distribution[16];
double AHS_num;
int AHSstart,AHSend;
int temp1,temp2, station;
double
dist1,dist2,traveldist,backdist,forwarddist,startdist,enddist,total_distance;
int start, finish, temp_a, temp_b, temp_c, temp_d;    //the station number of
starting/ending station
double dist;      //the distance of the truck will travel

double matrix[MAXLEN][MAXLEN], matrix2[MAXLEN][MAXLEN],
matrix3[MAXLEN][MAXLEN];
double temp_time, total_time, scale_factor, fuel_saving, total_fuel_cost,
unit_fuel_cost, count, AHS_labor;

main(int argc, char *argv[]){
      FILE *ifp, *ofp;
      int i, j, k, l;
      int station_num;
      char c, s[MAXLEN];
      double temp_num, real_num, debug, speed, customer_demand,
total_before_round, total_after_round, headway;

      ifp = fopen(argv[1], "r");
      ofp = fopen(argv[2], "w");

      printf("ok\n");


      c = getc(ifp);
      while(c != EOF){
            // skip the spaces or TAB or the new line char,
            //until next number read
            while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
            s[1] = '\0';
            i = 0;
            /* get the number of stations */
            if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                  while(isdigit(s[++i]=c=getc(ifp)))
```

```
                ;
            s[i] = '\0';
            station_num = atoi(s);  //convert the array into the integer
value

      for(i=0;i<station_num;i++){
            truck_length[i] = 0;
            traffic_length[i] = 0;
            for(j=0;j<station_num;j++){
                  matrix[i][j] = 0;
                  matrix2[i][j] = 0;
                  matrix3[i][j] = 0;
            }
      }

      for(i=0;i<16;i++){
            distribution[i] = 0;
      }

            while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
            s[1] = '\0';
            i = 0;
            /* get the scale-up factor */
            if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                  while(isdigit(s[++i]=c=getc(ifp)))
                  ;
            if(c=='.')//collect the fractional part numbers
                  while(isdigit(s[++i]=c=getc(ifp)))
                        ;
            s[i] = '\0';
            scale_factor = atof(s); //convert the array into the integer
value


            for(k=0;k<(station_num-1);k++){
                  while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
                  s[1] = '\0';
                  i = 0;
                  /* get the distance between each station */
                  if(isdigit(c))    //collect the integer digits of each
number
                        //put them in an array
                        while(isdigit(s[++i]=c=getc(ifp)))
                        ;
                  if(c=='.')//collect the fractional part numbers
                        while(isdigit(s[++i]=c=getc(ifp)))
                        ;
                  s[i] = '\0';
                  distance[k] = atof(s);  //convert the array into the real
number value
            }


            while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
            s[1] = '\0';
```

```
            i = 0;
            /* get the speed */
            if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                  while(isdigit(s[++i]=c=getc(ifp)))
                  ;
            if(c=='.')//collect the fractional part numbers
                  while(isdigit(s[++i]=c=getc(ifp)))
                        ;
            s[i] = '\0';
            speed = atof(s);  //convert the array into the double value



            while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
            s[1] = '\0';
            i = 0;
            /* get the headway */
            if(isdigit(c))    //collect the integer digits of each number
                        //put them in an array
                  while(isdigit(s[++i]=c=getc(ifp)))
                  ;
            s[i] = '\0';
            headway = atoi(s)/scale_factor;     //convert the array into the
integer
value


            for(k=0;k<16;k++){
                  while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
                  s[1] = '\0';
                  i = 0;
                  /* get the AHS entrance/exit stations */
                  if(isdigit(c))    //collect the integer digits of each
number
                        //put them in an array
                        while(isdigit(s[++i]=c=getc(ifp)))
                        ;
                  s[i] = '\0';
                  AHSexitnum[k] = atoi(s)-1;    //convert the array into the
real number
value
                  //printf("ahsnum:%d\n", AHSexitnum[k]);
            }


            while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                  c=fgetc(ifp);
            s[1] = '\0';

            printf("ok\n");

            /* read in the matrix */
            total_before_round = 0;
            total_after_round = 0;
            for(i=0; i<(station_num-1); i++){
                              for(j=(i+1); j<station_num; j++){
```

```
                        // skip the spaces or TAB or the new line char,
                        //until next number read
                        while((s[0]=c) == ' ' || c == '\t' || c == '\n')
                              c=fgetc(ifp);
                        s[1] = '\0';

                        k = 0;
                        /* get the numbers */
                        if(isdigit(c))    //collect the integer digits of
each number
                                    //put them in an array
                              while(isdigit(s[++k]=c=getc(ifp)))
                              ;
                        if(c=='.')//collect the fractional part numbers
                              while(isdigit(s[++k]=c=getc(ifp)))
                              ;

                        s[k]= '\0';
                        real_num = atof(s)*scale_factor;
                        total_before_round = total_before_round + real_num;
                        temp_num = floor(real_num);
                        if((real_num-temp_num)>0.50)
                              matrix[i][j] = temp_num + 1;
                        else
                              matrix[i][j] = temp_num;
                              total_after_round = total_after_round +
matrix[i][j];
                              }
                        }



            if((c=getc(ifp)) == EOF)
                  break;

      }//end of while(c!=EOF)

      printf("ok\n");
      total_time = 0;
      AHS_num = 0;
      for(i=0; i<(station_num-1); i++){
            dist = 0.0;
            for(j=(i+1); j<station_num; j++){
                  dist = dist + distance[j-1];
                  //take the AHS truck lane or not
                  if(dist > AHS_DIST){
                        AHS_num = AHS_num + matrix[i][j];
                        //printf("%d\t%d\n", i, j);
                        start = i;
                        finish = j;
                        //suppose the array AHSexitnum[] stores the exit
number of truck
entrance,i.e, AHSexitnum[0] = 5, means that the 1st
                        // AHS exit is exit num 5 on the freeway
                        //find out the closest AHS exit number to the
starting station
                        k=0;
                        while(AHSexitnum[k] <= start)
                              k++;
```

```
                        temp1 = AHSexitnum[k-1];
                        temp2 = AHSexitnum[k];
                        dist1 = 0;
                        dist2 = 0;
                        for(k=temp1;k<start;k++)            //the distance
from the entry point of the
truck on the freeway to the previous AHS access point
                              dist1 = dist1 + distance[k];
                        backdist = dist1;
                        for(k=temp1;k<temp2;k++)            //the distance
between the previous and the
next AHS access points
                              dist2 = dist2 + distance[k];
                        forwarddist = dist2 - dist1;

                        if((dist1/dist2) < BACKTHRESHOLD){        //the truck
shall travel backwards
                        AHSstart = temp1;
                        temp_a = i;
                        startdist = backdist;
                        }
                        else{
                        AHSstart = temp2;
                        temp_a = temp2;
                        startdist = forwarddist;
                        }

                        //find out the closest AHS exit number to the ending
station
                        k=0;
                        while(k<16 && AHSexitnum[k] < finish){
                              //printf("ssss k: %d\tahs exit: %d\tfinish:
%d\n",k,AHSexitnum[k],finish);
                              k++;
                        }
                        temp1 = AHSexitnum[k-1];
                        temp2 = AHSexitnum[k];
                        //printf(" k: %d\tahs exit: (%d %d)\tfinish:
%d\n",k,temp1,temp2,finish);

                        dist1 = 0;
                        dist2 = 0;
                        for(k=finish;k<temp2;k++)           //the distance
between the freeway
destination of a truck and the first AHS egress point beyond that
destination
                              dist1 = dist1 + distance[k];
                        //backdist = dist1;
                        forwarddist = dist1;
                        for(k=temp1;k<temp2;k++)            //the distance
between that AHS egress point
and the AHS egress point immediately before it
                              dist2 = dist2 + distance[k];
                        //forwarddist = dist2-dist1;
                        backdist = dist2 - dist1;
                        if((dist1/dist2) < BACKTHRESHOLD){        //the truck
shall travel beyond
farther
                              AHSend = temp2;
```

```
                        temp_b = j;
                        enddist = forwarddist;
                  }
                  else{
                        AHSend = temp1;
                        temp_b = temp1;
                        enddist = backdist;
                  }
                  matrix2[temp_a][temp_b] = matrix2[temp_a][temp_b] +
matrix[i][j];
                  matrix3[AHSstart][AHSend] = matrix3[AHSstart][AHSend]
+ matrix[i][j];
                  //printf("ssss start end: (%d
%d)\n",AHSstart,AHSend);

                  //calculate the distance the truck will really travel
                  traveldist = 0;
                  for(k=AHSstart;k<AHSend;k++)        //the distance
the truck travel between
the two AHS access points
                              traveldist = traveldist + distance[k];
                  //printf("distance\t%d\t%d\t%4.1f\n",
AHSstart,AHSend,traveldist);
                  total_time = total_time +
(traveldist/speed)*matrix[i][j] +
((startdist+enddist)/speed)*matrix[i][j];
            }
      }
}


for(i=0; i<(station_num-1); i++){
      dist = 0.0;
      for(j=(i+1); j<station_num; j++){
            dist = dist + distance[j-1];
            if(0<dist&&dist<=50)
                  distribution[0] = distribution[0] + matrix[i][j];
            else if(50<dist&&dist<=100)
                  distribution[1] = distribution[1] + matrix[i][j];
            else if(100<dist&&dist<=150)
                  distribution[2] = distribution[2] + matrix[i][j];
            else if(150<dist&&dist<=200)
                  distribution[3] = distribution[3] + matrix[i][j];
            else if(200<dist&&dist<=250)
                  distribution[4] = distribution[4] + matrix[i][j];
            else if(250<dist&&dist<=300)
                  distribution[5] = distribution[5] + matrix[i][j];
            else if(300<dist&&dist<=350)
                  distribution[6] = distribution[6] + matrix[i][j];
            else if(350<dist&&dist<=400)
                  distribution[7] = distribution[7] + matrix[i][j];
            else if(400<dist&&dist<=450)
                  distribution[8] = distribution[8] + matrix[i][j];
            else if(450<dist&&dist<=500)
                  distribution[9] = distribution[9] + matrix[i][j];
            else if(500<dist&&dist<=550)
                  distribution[10] = distribution[10] + matrix[i][j];
            else if(550<dist&&dist<=600)
                  distribution[11] = distribution[11] + matrix[i][j];
```

```
                 else if(600<dist&&dist<=650)
                       distribution[12] = distribution[12] + matrix[i][j];
                 else if(650<dist&&dist<=700)
                       distribution[13] = distribution[13] + matrix[i][j];
                 else if(700<dist&&dist<=750)
                       distribution[14] = distribution[14] + matrix[i][j];
                 else if(750<dist&&dist<=800)
                       distribution[15] = distribution[15] + matrix[i][j];
           }
     }

       /* truck length */
       total_distance = 0;
     for(i=0; i<(station_num-1); i++){
           total_distance = total_distance + distance[i];
           for(j=0; j<=i; j++){
                 for(k=(i+1); k<station_num; k++){
                       truck_length[i] = truck_length[i] + matrix2[j][k];
                       traffic_length[i] = traffic_length[i] +
matrix3[j][k];
                 }
           }
     }
     printf("total distance\t%4.1f\n", total_distance);

       /* fuel saving */
       fuel_saving = 0;
       total_fuel_cost = 0;
       unit_fuel_cost = 10;
       for(i=0; i<(station_num-1); i++){
           total_fuel_cost = total_fuel_cost +
distance[i]/unit_fuel_cost*(traffic_length[i]+1);
           fuel_saving = fuel_saving +
traffic_length[i]*0.1/unit_fuel_cost*distance[i];
       }



    /* print out the number of stations */
      fprintf(ofp, "Number of stations\t%d\n", station_num);

      /* print out the running time and headway */
      fprintf(ofp, "Speed and headway\t%4.2f\t%4.2f\t\n", speed, headway);

      /* print out the distance bwtween 2 stations */
      fprintf(ofp, "Distance between each station\t");
      for(i=0;i<(station_num-1);i++){
            fprintf(ofp, "\t%4.1f", distance[i]);
      }
      fprintf(ofp, "\n");



      fprintf(ofp, "The number of trucks which are travelling in AHS lane
between
each station during 1 day\t\t\t");
      for(i=0; i<(station_num-1); i++){
            fprintf(ofp, "\t%d", traffic_length[i]);
      }
```

```
	fprintf(ofp, "\n");

	fprintf(ofp, "The number of trucks which are attracted to AHS lane 
between 
each station during 1 day\t\t\t");
	for(i=0; i<(station_num-1); i++){
		fprintf(ofp, "\t%d", truck_length[i]);
	}
	fprintf(ofp, "\n");

	fprintf(ofp, "Hourly AHS truck length\t\t\t");
	for(i=0; i<(station_num-1); i++){
		fprintf(ofp, "\t%4.1f", traffic_length[i]/(24*(60/headway)));
	}
	fprintf(ofp, "\n");


	/* print out the matrix */
	fprintf(ofp, "The OD Matrix\n");
	for(i=0; i<station_num; i++){
		for(j=0; j<station_num; j++)
			fprintf(ofp, "\t%4.1f", matrix[i][j]);
		fprintf(ofp, "\n");
	}

	//fprintf(ofp, "The OD Matrix\n");
	//for(i=0; i<station_num; i++){
	//	for(j=0; j<station_num; j++)
	//		fprintf(ofp, "\t%4.1f", matrix2[i][j]);
	//	fprintf(ofp, "\n");
	//}

	fprintf(ofp, "Daily OD demand and Rounded hourly OD 
demand\t\t\t%4.1f\t\t\t%4.1f\n", total_before_round, total_after_round);
	fprintf(ofp, "\n");

	/* print out AHS truck's total travelling time */
	fprintf(ofp, "The Daily total AHS and truck travelling time 
is\t%4.1f\t%4.1f\n", (total_time+AHS_num*(headway/2 + 15)), 
total_time*1.25);
	fprintf(ofp, "\n");
	fprintf(ofp, "The number of trucks which take AHS truck 
lane\t\t%4.1f\n", 
AHS_num);
	fprintf(ofp, "Total fuel cost\t\t%4.1f\n", (total_fuel_cost-
fuel_saving));
	fprintf(ofp, "Total fuel saving\t\t%4.1f\n", fuel_saving);


	AHS_labor = total_distance/speed*24*(60/headway);
	fprintf(ofp, "Daily AHS and truck labor time\t\t%4.1f\t%4.1f\n", 
AHS_labor, 
AHS_labor*1.25);

	fprintf(ofp, "The distribution of travel distance\t\t\t");
	for(i=0; i<16; i++){
		fprintf(ofp, "\t%d", distribution[i]);
	}
	fprintf(ofp, "\n");
```

```
        return 0;
}
```

ACKNOWLEDGEMENT

This report is prepared in cooperation with the State of California, Business, Transportation and Housing Agency, Department of Transportation, and Partners for Advanced Transit and Highways (PATH). The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

The authors of this report would like to thank the California Department of Transportation (Caltrans) and the PATH Program for funding the research under the California Department of Transportation's Interagency Agreement No. 564180, Task Order No. 4236. Continuous support of Pete Hansra and Hassan Aboukhadijeh of Caltrans Division of Research and Innovation (DRI) are gratefully acknowledged. Dr. Mohamed AlKadri also of Caltrans Division of Research and Innovation provided detailed configuration data regarding the California portion of the Interstate 5. Dr. Randolph Hall of University of Southern California provided the traffic data regarding the same freeway. The authors would also like to thank them for their assistance.