Combining Retrospective Optimization and Gradient Search for Supply Chain Optimization

by

Stewart Liu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Philip M. Kaminsky, Chair
Professor Zuojun Max Shen
Professor Carlos Daganzo

Summer 2017

Combining Retrospective Optimization and Gradient Search for Supply Chain Optimization

Abstract

Combining Retrospective Optimization and Gradient Search for Supply Chain Optimization

by

Stewart Liu

Doctor of Philosophy in Engineering - Industrial Engineering and Operations Research

University of California, Berkeley

Professor Philip M. Kaminsky, Chair

In initial work, we found a version of Retrospective Optimization, in which we optimize over a single randomly generated long sample path, is often effective for optimizing policy parameters in relatively simple stochastic supply chains. In these applications, the optimization problem is frequently an integer program. However, preliminary efforts to directly extend this methodology to more complex supply chains, and to optimize risk mitigation strategies, were in many cases too slow to be effective.

To address this limitation, we first develop a two-stage algorithm that uses Retrospective Optimization over a relatively short time horizon to provide starting points for stochastic approximation gradient search. We perform extensive computational experiments to compare this approach to Retrospective Optimization without gradient search on a sequence of increasingly complex supply chains.

In a data-driven setting where the policy parameters are set using available past data rather than a randomly generated sample path, the resulting MILP formulation presents similar computational challenges to those found in our Retrospective Optimization approaches. This observation motivates us to modify and test our algorithm in a data-driven setting, using sales data from a major European grocery chain store. We focus on a setting in which policy paramters can be a function of exogenous factors such as the day of the week and the outside temperature. We examine complex inventory management models that involve perishable inventory. and show that with suitable modifications, the same two-stage algorithm is effective for determining the data-driven inventory policy parameters.

To Mom and Dad

Contents

List of Figures

List of Tables

Acknowledgments

The journey towards a Ph.D. degree has been an extraordinary one. Words alone can never describe the enormous challenges, both academically and emotionally, that stood between me and my goal. I would never have been able to overcome those challenges without the continuous love and support from everyone in my life.

First, I would like to extend my greatest gratitude to my advisor, Professor Phil Kaminsky, who tirelessly guided me, both intellectually and spiritually, through this otherwise impossible journey. I cannot thank him enough for his support and patience over these years.

I would also like to thank my thesis committee, Professors Max Shen and Carlos Daganzo, as well as Professor Lee Schruben, for their support. Their invaluable feedback over the years helped shape my thesis into what it is today.

Most importantly, I want to thank my mom and dad, who provided me with the unconditional love and support that I can never repay. Without them paving the road before me, I would not be in graduate school, or even in America today. Mom and Dad, *wo ai ni men*, forever and always.

To my extended family members: Grandpa Joseph and Grandma Ping, Thank you for cheering for me along the way. Grandpa Bin-Hsun, Grandma Bi-Yun, thank you for watching over me from above, may you rest in peace. My uncles, aunts, and cousins around the world, thank you for all your words of encouragement over the years of my life.

To all my loving friends: Dennis, Anran, Doris, Nancy, Joyce, Serena, Casey, Sofia, Michael, Moon, Wayne, Nicole, Jenny, Anqi, Lily, Audris, Sharon, Yichia, Nguyen, Ed, Phoebe, KC, Cheng, Dan, Shiman, Amber, Erik, Kevin, and many more that I did not mention, thank you all for always providing me with ears to listen and a shoulder to cry on.

Last but not least, a special thank you to my beloved Ying, for standing by my side in my darkest hours and being the brightest star in my life.

Xie-xie

Chapter 1

Introduction

Over the past several years, we studied supply chain systems where operational parameters can be set by formulating and solving mixed integer linear programming (MILP) models. Our research is motivated by two real world problems in the fields of biopharmaceutical manufacturing and grocery store inventory management. While these problems seem very different on the surface, the modeling and solution techniques used in both of these problems present similar challenges. Effective operational parameter settings for both of these types of supply chains can be obtained by solving MILP's that model the supply chain systems over a duration of time. These MILP's are often large and require too much time to solve to be practical. We propose and develop an approximate solution algorithm that utilizes a combination of traditional MILP solver and gradient search to significantly reduce solution times to these MILP's while providing a solution reasonably close to that of the complex MILP.

In the biopharmaceutical industry, firms often face the need to quickly and efficiently optimize supply chain operating parameters in complex, stochastic supply chains. It turns out that many of these problems share a specific structure: *given a sample path of realizations of the random variables in the problem over some finite horizon*, optimal policy parameters *for that specific sample path and that horizon* can be found by solving a (potentially very large) MILP. Given this insight, we have developed an effective two-step heuristic approach for finding effective operating parameters for these large stochastic problems, which we call ROGS, based on a hybrid of integer-programming-based *Retrospective Optimization* and *Stochastic Gradient Search*. Although these two techniques have separately (and in a few cases, together) been the focus of much research over the past decades, this is the first work, as far as we know, to combine these algorithms with simulation in a way that is flexible, relatively easy to implement in many settings, and applicable to optimizing the complicated and generally non-convex models characterized above. Indeed we have had success using our approach to quickly and efficiently optimize supply chain policy parameters in a number of settings for biopharmaceutical firms that are members of the NSF-supported Center for Excellence in Logistics and Distribution.

Firms in the biopharmaceutical industry have supply chains that involve multiple stochastic elements including random demand, yields, fermentation times, filtration times, quality control times, etc. In addition, these supply chains are subject to disruptions due to natural disasters and human errors. There are typically a large number of supply chain operating policy parameters to optimize in these supply chains, where, for instance, an order-up-to policy might dictate raising inventory to y at a specific site each period, or a periodic shipping policy might dictate shipping available inventory from one site to another every x days. Optimizing policy parameters (such as the x or y in the examples above) in this sort of complex, stochastic environment typically requires complicated mathematical models with solution approaches that are very specific to the exact structure of the model (particularly when a large number of system parameters need to be determined simultaneously.) Much to our surprise, Retrospective Optimization via integer programming proved effective for solving these real-world problems.

Retrospective Optimization [17] is a type of sample path optimization [29] commonly used for finding solutions to stochastic optimization problems. This approach is inspired by retrospective analysis, where managerial decisions are made through an exploration of what *would have been* the best decisions in the past based on subsequently realized data and performance[17]. To apply Retrospective Optimization, we simulate a potential future, and then determine the retrospective solution for this projected future. Specifically, given distributions for a problem's random variables, one possible realization for all variables is generated, and the objective function is optimized given this now-deterministic sample path. In other words, we are solving the following problem: If we know the realization of future random variables, what is the most effective policy? In "retrospective" terms, suppose that we are at some point in the future having seen the value that the random system parameters have taken on. Given this scenario, what *would have been* the best policy to implement?

For supply chain optimization problems where the underlying distribution of the random variables are unknown to the decision maker, generating a sample path of random variables is challenging. Instead of attempting to fit a distribution to past data, it is possible to directly optimize supply chain paramters using this past data. This approach is known as Data-Driven Optimization. When the random variables in these models are correlated with some observable data, a state-dependent policy can be implemented. In place of a set of stationary parameters, such as order-up-to levels, a state-dependent policy allows parameters to be a function of exogenous variables, such as temperature of the day or the day of the week. The goal then becomes to determine an effective formula for day-to-day operational parameters. To obtain effective coefficients this type of (typically linear)function, a highlycomplex

MILP is formulated. The resulting MILP is very similar to those found in Retrospective Optimization and presents similar challenges with lengthy computation times.

For the types of supply chain policy parameter optimization problems we consider where it is reasonable to assume a linear cost function, given a sample path, parameters can typically be optimized using a mixed-integer program, where integer variables are necessary to model the implementation of *policies*. Recall we are optimizing policy parameters, and not decisions, and consider the example of setting inventory policy in a simple inventory system. If future demand is known and our objective is to minimize inventory costs by determining shipment quantities in each period, shipments would equal future demand, and solving this class of models would require solving linear programs in many cases. However, this approach would not give insight into how to actually manage the supply chain, because in reality, future demand is not known. However, if stationary policy parameters (such as base-stock levels for example) are optimized, the resulting mathematical program is a MILP, and the resulting policy is *implementable* because it doesn't directly depend on specific demand realizations.

To illustrate this, consider the example of a simple single stage k-period finite horizon periodic review inventory model with deterministic lead time and random demand. Assuming no loss sales (i.e. backorder is allowed with a penalty cost), such system can be stochastically optimized under a basestock policy. However, with a demand sample path, d_t in period $t = 1, 2, 3, ..., k$, this system operating under the basestock policy can also be modeled as a retrospective MILP.

We let π be the backorder cost and h be the unit holding cost. Let the variable s represent the basestock level, which we also assume to be the inventory level at the beginning of time. Let I_t be inventory at time t, T_t be demand met from inventory at time t, B_t be backorder at time t, and s be the stationary base stock level. Given a vector of demand $\bar{X}_k = [d_1, d_2, ..., d_k]$, the optimal basestock level, s^*, for that vector of demand can be found by solving the following integer program:

$P1(\bar{X}_k)$:

$$\begin{aligned}
min \quad & \frac{1}{k}\sum_{t=1}^{k}\Big[\pi B_t + hI_t\Big] & & \\
s.t. \quad & T_t = min\{s, d_t + B_{t-1}\} & t = 1, 2, ..., k \qquad & (1.1)\\
& B_t = d_t + B_{t-1} - T_t & t = 1, 2, ..., k \qquad & (1.2)\\
& I_t = s - T_t & t = 1, 2, ..., k \qquad & (1.3)\\
& T_0 = 0, B_0 = 0, & & (1.4)\\
& T_t \geq 0, B_t \geq 0, I_t \geq 0 & t = 1, 2, ..., k \qquad & (1.5)
\end{aligned}$$

Note that the minimization constraint (1.1) can equivalently be formulated as a set of inequalities using a binary variable and a sufficiently large constant. (We detail this in

Section 3.2.)

In the specific projects that motivated this work, our goal was to optimize stationary policy parameters in what is effectively an infinite horizon setting. We did this using Retrospective Optimization by solving the integer program resulting from generating a "sufficiently long" sample path. While we were able to generate effective policies by solving these integer programs for very long time horizons (many years of daily decision-making), these integer programs take a long time to solve (sometimes days), making sensitivity analysis challenging. To reduce solution times, we have had success combining this integer-programming-based Retrospective Optimization approach with an approach based on Stochastic Gradient Estimation and Search. In our supply chain problems, parameter decision variables take on continuous values. Thus, we attempt to optimize these parameters by assigning initial values, estimating gradients, and searching in improving directions. This approach can be used to obtain an approximation of a local optimum [10]. However, the cost functions of supply chain problems such as the ones we are dealing with are in general not convex functions of policy parameter settings. Thus, we are motivated to utilize a two-stage approach for finding effective parameter settings: first, we solve a relatively computationally inexpensive Retrospective Optimization integer program to find a "ballpark solution" – a solution that we hope is sufficiently close to the optimal one that local search will prove effective. Then, we use this solution as the starting point for the gradient search. For the examples we have tried, this approach is be nearly as effective as solving the large, complex MILP's described above, but takes orders of magnitude less time, thus allowing us to apply this approach to even more complex supply chains, and to more easily support "what-if" analysis.

We extend the application of our algorithm further by studying supply chain systems that deal with perishable inventory, such as blood bank or food items. Using data obtained from a major European grocery chain store, we build state-dependent data-driven models with perishable inventory that include a large number of integer variables and constraints. We then apply our algorithm to these data-driven models and observe the results.

While there are no doubt other settings where retrospective optimization problems can be expressed as MILPs, and our approach may also work for those, we developed our approach of combining Retrospective Optimization MILP's with Stochastic Gradient Search, which we call ROGS, in the context of supply chain problems, and in this paper we explore this approach. Specifically, we present a series of computational experiments developed to better understand the performance of ROGS, and to refine the details of its implementation. In the next section, we review relevant literature. In Section 3.1, we present our two-step ROGS approach in more detail. In Section 3.2, we present experiments designed to validate the performance and accuracy of ROGS under different operational parameters. In Section 4.3, we discuss a minor modification to our algorithm to adapt to the state-dependent data-driven models discussed in Section 4.2. Finally, Section 4.5 presents the results of these applications.

Chapter 2

Literature Review

There are a variety of relevant streams of literature in simulation optimization and supply chain inventory optimization. Below, we consider approaches for optimizing stochastic objective functions based on gradient estimation and search algorithms, as well as Retrospective Optimization. Because both our motivating problems and computational testing focuses on supply chain inventory position, we also explore the supply chain inventory positioning literature in both traditional basestock policies and data-driven policies.

2.1 Stochastic Gradient Estimation

In a typical constrained stochastic optimization problem, the objective is either directly or indirectly a function of both random variables and decision variables. We seek to optimize the decision variables, but must account for random problem parameters. Clearly, given a set of decision variables, we can obtain an estimate of the objective function value by sampling one or more instances of the random variables.

A traditional approach for solving this type of problem involves estimating the gradient of the stochastic objective function, and searching in improving directions until the objective function value improves by less than a predetermined threshold. A simple way to estimate the gradient involves computing the differences in the objective in each individual search direction. This approach is known as the finite difference approximation, in which the gradient is estimated in direction or variable at a time while keeping all other variables fixed. The gradient estimation involves perturbing each variable by a small amount and observing the change in objective function value. One of the most well-known algorithms of this type is the Kiefer-Wolfowitz stochastic approximation algorithm [20]. This algorithm begins with an initial step size $c_0 = c$ to estimate the gradient in each direction, then uses gradient estimate step sizes

$c_n = cn^{-1/6}$ for all directions at iteration n. It then searches in the direction with the steepest gradient estimate using an initial search step size of $a_0 = a$, with $a_n = an^{-1}$ for iteration n. This algorithm has been shown to have an asymptotic convergence rate of $\frac{1}{\sqrt{n}}$ [10]. In practical implementations, the choice of the initial step size is crucial, but to the best of our knowledge, there is no general methodical approach for determining step size. Nevertheless, with proper step-size choices, the Kiefer-Wolfowitz algorithm works quite well in various applications[4][10].

While the Kiefer-Wolfowitz algorithm explores one decision variable at a time, a method known as simultaneous perturbations [10] explores a combination of multiple directions in one single iteration. The benefit of this approach is obvious – it can potentially take fewer iterations to find local optimum. However, the computational burden of this algorithm is significantly higher than the finite differences approach, so this approach is highly impractical without effective problem-specific approaches for effectively selecting the search direction. Without such a problem-specific algorithm to reduce the number of combinations to explore in each iteration, this procedure is likely to be inferior to the finite difference approaches if the gradient estimation is computationally relatively inexpensive [10].

Regardless of the search algorithm, there are still two major issues with this set of approaches. First, as with deterministic methods,the global optimum is challenging (or impossible) to find if the objective function is not convex. In general, only a local optimal solution can be obtained with gradient search, so the effectiveness of the algorithm is a function of the starting point of the search. Heuristically, it is helpful to have a starting point "close enough" by some measure to the global optimum, so that the search can find an optimal or near-optimal point. The second issue lies with stochastic gradient estimation. If the objective function has high variance, as the gradient estimation step size gets smaller, the difference in the objective function value can be extremely sensitive to random noise. We address the first issue by using a Retrospective Optimization integer program to find an effective starting point. For the second, we note that we have observed for our application that while the objective function value varies widely, the optimal decision variables do not vary as as simulated sample paths change. We therefore eliminate the noise of gradient estimation by performing the sample path simulation prior to the start of the gradient search and fixing all random variables at their realization from then on.

2.2 Retrospective Optimization

Another class of approaches for optimizing stochastic objective functions is known as Sample Path Optimization. This class of approaches involves using one or more samples of the random variables in the problem to determine the objective function values.

Once the samples are generated via simulation, the problem becomes deterministic, so a wide variety of deterministic optimization tools can be employed. The most common of these approaches is known as Sample Average Approximation (SAA). To minimize an expected value objective function, SAA approaches first randomly generate a number of possible realizations of the random parameters of the stochastic function, and then optimize the average objective function value over the generated samples – a deterministic objective function. If the objective function of each sample is differentiable and its derivative converges to that of the original objective function, the optimal solution to the SAA problem converges to that of the original stochastic problem, as the number of sample paths approaches infinity[21]. SAA has been used to solve various stochastic optimization problems in areas such as electric power systems [24], telecommunication networks [24], financial planning [37], budget allocation [30], vehicle allocation and routing [36], and of course inventory control [37]. Further, if there is a unique optimal solution to the original stochastic problem, then the unique optimizer also converges to that of the stochastic problem [21]. The SAA approach often works well with finite horizon planning problems, or problems with a steady state behavior easily captured by a relatively short finite sample path, but it is less clear how effective this approach might be with infinite horizon problems, or problems with low likelihood but significant random events such as disruptions. In these cases, computational resources may be better utilized in simulating one relatively long sample path instead.

For stochastic integer programming problems with long planning horizons, Retrospective Optimization is an approach that, in contrast to SAA, utilizes a single (long) sample path instead of multiple sample paths. Healy and Schruben [17] propose the generation of this single long sample path, followed by the optimization of the resulting large deterministic optimization problem based on this sample path, an approach that works well in the settings that interest us, where supply chain status is periodically reviewed, and policies that are a function of this status are applied. Assuming that random variables are i.i.d. from period to period, as the horizon length approaches infinity, the per-period cost of the finite horizon problem converges almost surely to the true per-period objective function value[29][27]. In addition, the optimizer for the finite horizon problem also converges to the optimizer for the original infinite horizon problem if the following conditions hold almost surely [29]:

1. For any horizon length $1 \leq n < \infty$, the n-period finite horizon objective function is lower semicontinuous and proper.
2. The n-period finite horizon objective function epiconverges to the infinite horizon objective function.
3. The infinite horizon objective function is proper and its set of minimizers is nonempty and compact.

These conditions are satisfied in our seting when demands and leadtimes are finite, a policy structure is specified, and a minimizer exists the infinite horizon problem. As we mentioned in the introduction, one of the advantages of our approach is that standard integer programming software can be used to solve our Retrospective Optimization problems. Unfortunately, the solution time of these integer programs increases dramatically as the planning horizon or problem complexity increases. To address this, our algorithm seeks to dramatically reduce solution times while employing the same general approach as Retrospective Optimization.

2.3 Hybrid Algorithms

In addition to sample path and gradient estimation approaches, researchers have considered hybrid algorithms that combine several approaches. Fu and Healy (1997) [11] consider a single stage inventory system under an (s, S) policy and outline the trade-offs between using gradient search and customized retrospective optimization algorithms to optimize system parameters. As they observe gradient search can be slow to converge, while retrospective optimization can be computationally expensive as the simulation horizon grows long. They propose a hybrid method that combines their customized retrospective optimization and gradient estimation approaches by iteratively simulating random demand for one or several periods at a time. For every N periods, they first use their custom retrospective algorithm to establish the order-up-to point, then they use gradient estimation to adjust the decision variable value once. This procedure is repeated until the end of the predetermined planning horizon. Bashyam and Fu (1998) [1] later apply a similar algorithm to an inventory system with random lead time under (s, S) policy. Here, they emphasize finding a good starting point for the gradient descent search algorithm. They used a two stage method that first obtains the optimal quantity from an EOQ relaxation and then performs line search via simulation to improve on the optimal Q. More recently, Tsai and Zheng (2013) [34] consider a two-echelon inventory system with service level constraints under a basestock policy. They use a two-step hybrid method that first relaxes the non-linear response time constraints, then iteratively add the non-linear constraints one by one until a feasible solution is obtained. These hybrid algorithms provide good results for the specific problems for which they are developed. However, it is unclear how these algorithms can be generalized to other problem settings. Indeed, to our knowledge, there are no general purpose hybrid algorithms or approaches flexible enough to be applied to a variety of general supply chain and inventory management problems without a great deal of problem-specific development.

2.4 Inventory Positioning

Recall that this research was motivated by inventory management challenges in biopharmaceutical supply chains. Indeed, there is a great deal of research focusing on inventory decisions in supply chains, but many existing approaches are designed for specific models and types of policies and hence lack the flexibility to quickly and efficiently optimize a wide range of real-world supply chains For obvious reasons, is often essential to incorporate randomness in the models. Many researchers model this randomness, but practically solvable models often limit the number or nature of these random variables. For instance, demand may be modeled as random, but lead times, which are often stochastic in the real world, are often deterministic in models.

Simpson (1958) [33] first investigated the notion of a serial supply chain by breaking down a supply chain operation into multiple "stages" and assuming that "in-process inventory" can be kept in each stage. Simpson models this serial supply chain under guaranteed lead time between stages, then optimizes it under a basestock policy for each stage. The model also assumes all inventory, regardless of its location, has the same average cost, and hence only seeks to minimize the units of inventory in the system as a whole. Simpson then adds bounded random demand into the model, and concludes that at each stage the optimal basestock level should either be zero or the maximum allowed under these assumptions.

Similar multi-echelon inventory systems were later investigated under the assumption of unbounded stochastic lead time or demand. Some of these models include a backorder penalty cost for cases of stockouts and seek to minimize the cost by adjusting inventory basestock level. A recursive cost function for a general N-stage serial supply chain with echelon holding costs is given by Clark and Scarf (1960) [7] and later revisited by Chen and Zheng (1994) [5]. The first stage (i.e. the downstream stage closest to demand) of this formulation is similar to the newvendor problem. However, optimizing the basestock levels in stages other than the first can be very computationally expensive and therefore impractical. A heuristic bound was developed by Shang and Song (2003) [32] using formulation similar to the newsvendor objective and shown to be accurate.

Other related work includes Federgruen and Zipkin's (1983) [8] investigation of an infinite horizon two-stage supply chain system with stochastic demand under periodic review. Here, they solve for optimal policy parameters, including order quantity and shipment quantity from the depot to the retailer, under a critical number policy. An approximate dynamic programming algorithm is proposed to approximate the optimal policy parameters. Zipkin later points out that stationary demand distributions are not a realistic assumptions as seasonal and other factors often affect demand. However, he noted that his algorithm has a smoothing effect on fluctuating demand and thus is applicable to real world systems. [38].

Inventory positioning models in more complex supply chains have been explored, but most are based on the guaranteed service level model. This class of problems assumes a 100% service level at each stage of the supply chain within a committed service time. This assumption directly implies an upper bound on the basestock levels at each stage. Expanding on Simpson's [33] serial supply chain system, Graves [15] observes that, under a guaranteed service level, this model can be formulated as a dynamic program. Further, Inderfurth [19] shows that the problem can be solved as a shortest path problem. Graves and Willems [16] later expand on the serial structure of the supply chain and study supply chains that can be modeled as a spanning tree. They propose a dynamic programming algorithm to solve for the optimal basestock levels[16]. Magnanti et al. [23] later propose an alternate integer programming approach to solve spanning tree supply chain systems. Although spanning trees can capture assembly, distribution, and other complex supply chain systems, this model assumes a guaranteed service level, and doesn't address general supply chain systems that include cycles. In contrast, our approach is applicable to any supply chain structure that can be modeled using a MILP as described above, and to systems without guaranteed service.

2.5 Data Driven Inventory Models

In most stochastic inventory models, the random demands are drawn from a known or a parametric family of distributions. However, a non-parametric data-driven approach can be used to model supply chain systems using past data. With this approach, operational parameter decisions are driven directly by past data instead of being based on estimated distributions. The distribution-free newsvendor model goes back to the pioneering work of Scarf[31] who is the first one to determine an inventory policy based on past demand observations instead of a theoretical demand distribution. Gallego (1993) [12] extends Scarf's work by adding a second purchasing opportunity, fixed ordering costs, random yields and capacity constraints. Bertsimas (2006) [2] proposes a new approach using robust linear or mixed-integer programming. They argue that their approach is both intuitively easy to understand and applicable in practice as well as outperforms dynamic programming solutions under certain circumstances.

Other research in this class of problems include the work of Godfrey and Powell (2001) [13] that uses the Concave, Adaptive Value Estimation algorithm that result in a nonlinear approximation to inventory optimization problems without the assumption of any underlying distribution of the random variables. Huh and Rusmevichientong (2009) [18] establishes a bound for the cost difference between inventory policies obtained using historical demand data versus a known underlying demand distribution. Levi et. al. (2007) [22] studies the number of samples required to guarantee certain bounds for the data-driven approach. Extensive literature reviews on operational statistics and data driven optimization can be found in Chu et.al. [6] and Ramamurthy et. al. [28].

In this class of optimization problems, inventory policies can be a function of external variables as well as the system state. For instance, policies may depend on the day of the week, or the outside temperature. One area where this approach seems to be particularly promising is day-to-day supply chain operations for grocery retailers handling perishable goods[3].

2.6 Perishable Inventory Models

Research on inventory policies for products with fixed lifetime and stochastic demand goes back to the pioneering works of VanZyl [35], Nahmias[25] and Fries[9]. Since then, numerous publications in this field have emerged. Nahmias[26] and Goyal[14] provide extensive review on theseof this literature. Nahmias[25] and Fries[9] show that the order quantity depends on the length of the planning horizon and the age structure of the inventory on-hand. Nahmias[25] finds that increasing the on-hand inventory leads to decreasing order quantities, but this effect is more significant for an increase of the on-hand inventory of new than those of old items. Fries[9] comes to a similar result. As opposed to Nahmias[25] who decides on the order quantity, the on-hand inventory is the decision variable in Fries[9]. The author distinguishes between products that expire at or after the planning horizon and those that expire before. In the former case perishability is not an issue and the target inventory does not depend on the so-called immmortal inventory. In the latter case, the order depends on the age distribution of the inventory. If the on-hand inventory consists mainly of new items that are to expire only in later periods, orders should be delayed since, depending on the demands in the earlier periods, additional inventory may not be required until later periods. If the on-hand inventory is to expire soon, orders should be placed to avoid stockouts. Fries[9] also find that if the planning horizon and the product lifetimes are greater than two periods, then neither the basestock nor the (s,S) policy is optimal.

Chapter 3

The Two-Step ROGS Algorithm

The Retrospective Optimization approach to optimize complex supply chain decisions can be flexible and effective. However, the challenge lies with the extremely long computation times required to solve these retrospective MILP's. To expedite the solution times, we propose a two-stage algorithm that combines the retrospective MILP models with gradient search techniques to obtain effective policy parameter within a reasonable time.

3.1 ROGS: The Two-Stage Algorithm for Hybrid Retrospective Optimization and Gradient Search

In general, we seek the policy parameters that minimize the expected operating cost of a supply chain:

$$E_X[f(\theta, X)]$$

where $\theta = (\theta_1, \theta_2, ..., \theta_l)$ is a l-dimensional vector representing the variables over which we are optimizing (such as basestock levels), and $X = (X_1, X_2, ...)$ is an infinite-dimensional vector capturing the randomness inherent in our infinite horizon supply chain problem.

To (approximately) minimize $E[f(\theta, X)]$, it suffices to optimize a long but finite horizon Retrospective Optimization Integer Program (ROIP) because of the convergence properties of both the average per-period cost function and the optimizers. We first generate a sufficiently large sample path of the random variables, X, (think of a long but finite set of demands, yields, etc.) representing one possible realization of a finite amount of the infinite supply chain horizon, “long enough” to capture the near steady state operation of the supply chain. Given this sample path, we obtain a deterministic estimator function of $E[f(\theta, X)]$, denoted $\widehat{f}(\theta)$. This ROIP can then be solved to optimize policy parameters for that sample path, with the hope that these will be effective policy parameters for the original problem.

However, for a long horizon, these ROIP's often consist of many integer variables and are very difficult to solve. Fortunately, for the type of supply chain planning problems mentioned above, evaluating the system-wide cost function via simulation is straightforward and computationally inexpensive given the policy parameters. Even with more complex supply chains, the total operating cost can typically be estimated using simulation techniques. Our two-step algorithm exploits this observation by utilizing the Kiefer-Wolfowitz stochastic approximation gradient search algorithm [20], which has a $\frac{1}{\sqrt{n}}$ asymptotic convergence rate [10]. In practice, we introduce both an iteration limit, N, and a threshold, δ, so that the algorithm terminates the objective function either doesn't improve by more than the threshold in any descent direction, or the iteration limit is reached. N and δ can be adjusted to fit any model and computer. In general, through experimentation, we have found that it is useful to set δ roughly equal to 0.001% of the initial objective function value, and N set such that the computation time does not exceed two hours.

There are two classes of approaches that could be employed for gradient estimation and search. One of these is to generate a new sample path for every gradient estimation and search step. In other words, $\widehat{f}(\theta)$ could be redefined with a new randomly generated sample path every time it is evaluated. This approach works well when there are few random variables, with low variance. For instance, for a simple basestock level setting problem in a four-stage serial supply chain with random demand and constant lead times, this approach works quite well. However, as the number of random variables and the variance of those random variables increases, this approach is less effective. As the demand variance increases or as the model incorporates random lead times, for example, the gradient search tends to terminate after just a few iterations, because the random noise in each sample path is too great. Inevitably, a sample path will be generated such that the cost to operate under that sample path is much higher than the one previously generated, so the algorithm terminates.

In preliminary experiments, we noticed that while the objective function value varies depending on the sample path, the optimal decision variables are considerably less variable. Thus, we utilize a simplified approach: we hold the function $\widehat{f}(\theta)$ constant in each gradient estimation and search step, perturbing only θ, the decision variables. Under this approach, we find the optimal decision variables for one sample path at a time. We can then repeat the experiment multiple times and report the average optimal solution obtained. Using this approach, at each step of the gradient search, the gradient estimator of direction i is then given by:

$$\frac{\widehat{f}(\theta + c_i e_i) - \widehat{f}(\theta - c_i e_i)}{2c_i}.$$

At iteration n, $c_n = cn^{-1/6}$ for all directions i, and the incremental step size towards the next proposed solution in the search is $a_n = an^{-1}$, where c and a parameters of the algorithm. To guarantee almost sure convergence, the sequence $\{a_n\}$ must satisfy the following limiting conditions [10]:

$$\sum_n a_n = \infty \text{ , and}$$
$$\sum_n a_n^2 < \infty.$$

Given the predetermined improvement threshold δ, the algorithm will terminate and return the decision variables, θ, if for all directions i:

$$\widehat{f}(\theta + a_i e_i) - \widehat{f}(\theta) \leq \delta$$
$$\widehat{f}(\theta - a_i e_i) - \widehat{f}(\theta) \leq \delta.$$

As mentioned above, we also impose a predetermined maximum iteration limit, N, so that the algorithm will terminate regardless of descent improvement in $\widehat{f}(\theta)$, if:

$$n > N$$

Instead of attempting to directly optimize the large finite horizon ROIP described above, we also create a relatively short horizon version of the model. We then use our two-step algorithm, ROGS, to find a good solution for $\widehat{f}(\theta)$.

In general, to solve an ROIP given a length k sample path $\bar{X}_k$, which we label $P(\bar{X}_k)$, we take the following steps:

Algorithm (ROGS):

- Step 0: Assign the parameter values of ROGS
    1. Determine a sample path length, k, long enough for the objective function value of $P(\bar{X}_k)$ to capture the long run average cost of the supply chain system.
    2. Determine a short sample path length, $k_0 << k$, where the ROIP $P(\bar{X}_{k_0})$ can be relatively easily solved used a straightforward application of commercial optimization software.

3. Determine initial gradient estimation and search step sizes, a and c.
4. Determine the gradient search termination threshold δ

- Step 1: Solve $P(\bar{X}_{k_0})$

1. Generate a short sample path, $\bar{X}_{k_0}$, and formulate an ROIP $P(\bar{X}_{k_0})$.
2. Solve the ROIP to find the optimal vector of decision variables, θ'.

- Step 2: Solve $P(\bar{X}_k)$ using Gradient Estimation and Search

1. Generate a longer sample path, $\bar{X}_k$, and formulate an ROIP $P(\bar{X}_k)$.
2. Use θ' as a starting point for finite difference gradient estimation and search. At each iteration, the gradient estimation and step sizes are reduced following the sequences $a_n = an^{-1}$ and $c_n = cn^{-1/6}$.
3. Repeat the search until the estimated gradient in all directions is less than δ. Return the decision variable vector, θ^*

The applications that motivate us require optimization of multiple parameters in highly complex supply chains, with randomness in demand, yield, processing times, quality control testing and release times, etc. To effectively set supple chain parameters, we need models that explicitly capture this randomness, and that are flexible enough to model alternative supply change structures easily, and the procedure outlined above allowed us to meet these goals. In the next section, we present a series of experiments designed to test our ROGS approach in supply chain settings with known solutions, before applying to an increasingly complex set of more challenging-to-solve problems.

3.2 Computational Experiments

In this section, we present the results of a series of computational experiments intended to explore the effectiveness of this approach, as well as the impact of various algorithm settings and implementation decisions. To do this, we first consider a simple model with a known solution – a multi-stage serial supply chain inventory system under a basestock policy and periodic review; we then increase the complexity of that model in several ways.

Serial Supply Chain with Deterministic Lead Time

Problem Statement

For our first set of experiments, we consider a standard infinite horizon four-stage serial supply chain with stochastic demand under periodic review governed by a simple base-stock policy. All other parameters are deterministic. As we discuss below, we can find optimal basestock levels in a reasonable amount of time. We can thus compare the solution found by our approach (which would typically be used in systems where finding optimal policy parameters is intractable) with the optimal solution.

For our formulation, we index the intermediate supply chain stages $i = 1, 2, ..., N$ where stage 1 is responsible for satisfying downstream demand and stage N orders from an external supplier (upstream).

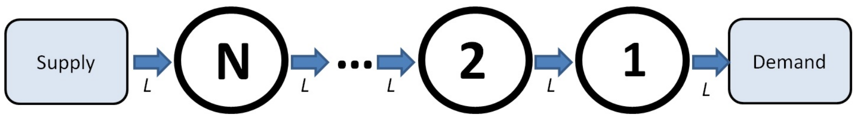


Figure 3.1: SSC-DL

We assume the external supplier will always be able to satisfy demand. We also index the discrete time periods forward in time, $t = 1, 2, ..., k$ where k is the length of the finite planning horizon. Using a retrospective optimization approach, we then generate a k-period demand sample path, $\bar{X}_k$, drawing random demand values from a given distribution (i.e. $\bar{X}_k = [d_1, d_2, ..., d_k]$). Given this indexing convention and sample path, we define the following parameters:

- d_t = demand in period $t = 1, 2, 3, ..., k$
- π = backorder cost (incurred in stage 4 only)
- h^i = holding cost at stage $i = 1, 2, ..., N$ of the supply chain
- L = lead time between stages

and the following variables:

- s^i = base stock at stage i (non-negative)
- I_t^i = inventory at time t at stage i
- T_t^i = inventory shipped at time t from stage i
- B_t^i = backorder at time t at stage i

As we discussed in the introduction, the resulting retrospective optimization problem is a MILP, which we formulate as follows:

SSC-DL($\bar{X}_k$):

$$
\begin{aligned}
&min \quad \widehat{f} = \frac{1}{k}\left[\sum_{t=1}^{k} \pi B_t^1 + \sum_{t=1}^{k}\sum_{i=1}^{N} h^i I_t^i\right] && && (3.1)\\
&s.t. \quad T_t^i = min\{I_{t-1}^i + T_{t-L}^{i+1}, d_t + B_{t-1}^i\} && i = 1,2,...,N \ , \quad t = 1,2,...,k && (3.2)\\
&\qquad B_t^i = d_t + B_{t-1}^i - T_t^i && i = 1,2,...,N \ , \quad t = 1,2,...,k && (3.3)\\
&\qquad I_t^i = I_{t-1}^i + T_{t-L}^{i+1} - T_t^i && i = 1,2,...,N \ , \quad t = 1,2,...,k && (3.4)\\
&\qquad T_t^{N+1} = d_t && t = 1,2,...,k && (3.5)\\
&\qquad T_0^i = 0 && i = 1,2,...,N && (3.6)\\
&\qquad B_0^i = 0 && i = 1,2,...,N && (3.7)\\
&\qquad I_0^i = s^i && i = 1,2,...,N && (3.8)\\
&\qquad T_0^i \geq 0 && i = 1,2,...,N && (3.9)\\
&\qquad B_0^i \geq 0 && i = 1,2,...,N && (3.10)\\
&\qquad I_t^i \geq 0 && i = 1,2,...,N \ , \quad t = 1,2,...,k && (3.11)
\end{aligned}
$$

Observe that although the objective function $\widehat{f}$ is written as a function of B and I, these variables are constrained by a series of equations and are ultimately a function of the basestock variables, s^i. Also, we use the standard approach to model a minimization in an integer program, introducing a binary variable, z, and a large constant, M, and replacing

$$T_t^i = min\{I_{t-1}^i + T_{t-L}^{i+1}, d_t + B_{t-1}^i\}$$

by the following set of inequalities:

$$
\begin{aligned}
I_{t-1}^i + T_{t-L}^{i+1} &\leq d_t + B_{t-1}^i + zM\\
d_t + B_{t-1}^i &\leq I_{t-1}^i + T_{t-L}^{i+1} + (1-z)M\\
T_t^i &\leq I_{t-1}^i + T_{t-L}^{i+1}\\
T_t^i &\leq d_t + B_{t-1}^i\\
T_t^i &\geq I_{t-1}^i + T_{t-L}^{i+1} - zM\\
T_t^i &\geq d_t + B_{t-1}^i - (1-z)M
\end{aligned}
$$

We in tests, we use our ROGS approach as described in Section 3.1.

- For a $k_0 << k$, we formulate a truncated version of the MILP: SSC-DL($\bar{X}_{k_0}$). As before, the formulation is the same except that we are using a truncated version of the same sample path, with indices $1..k_0$. We solve SSC-DL($\bar{X}_{k_0}$) using commercial solver CPLEX.

- To solve the full k-period problem, SSC-DL($\bar{X}_k$), we then:
  - Use the optimal basestock level obtained by SSC-DL($\bar{X}_{k_0}$) as the starting point
  - Using approximate gradient search to improve the total cost for the k-period planning horizon by changing the policy parameters one basestock level at a time. We let the initial gradient estimation and search step sizes, a and c, be about $5 - 10\%$ of the starting decision variable values.
  - Terminate the search when total cost improvement is below the threashold δ, or when search exceeds the maximum number of iterations, N, where
    - $\delta = 0.001\%$ of the initial objective function value.
    - $N = 2000$, roughly limits ROGS to 2 hours of computation time.

The Optimal Policy

Recall that we selected this problem setting to test our heuristic because we know how to find the optimal solution. Specifically, a recursive cost function for a general N-stage serial supply chain with echelon holding costs is defined by Clark and Scarf (1960) [7] as follows:

First, let $\underline{C}_0(x) = (p + h'_1)x^-$, and for $j = 1, ..., N$, let:

$$\begin{aligned} \hat{C}_j(x) &= h_j x + \underline{C}_{j-1}(x) \\ C_j(y) &= E[\hat{C}_j(y - D_j)] \\ s_j^* &= \text{argmin}\{C_j(y)\} \\ \underline{C}_j(x) &= C_j(min\{s_j^*, x\}) \end{aligned}$$

Then $s^* = (s_1^*, s_2^*, ..., s_N^*)$ is the optimal base-stock level for this system.

Experimental Design

For our experiments, we generate our retrospective optimization sample path of periodic demand with various distributions, including normal and beta distributions. The computational results can be found in Appendix A. In the following sections, we will discuss some select statistics from our observations. Primarily, we will look at the cases where demand follows a normal distribution with a mean of 32 and a standard deviation of σ. We let the lead time between stages be 1 ($L = 1$), use unit incremental holding costs ($h_1 = 4, h_2 = 3, h_3 = 2, h_4 = 1$), and set the backorder penalty cost of 36 per unit ($\pi = 36$). We then take this sample path and formulate the integer program as described in the previous section to solve for the optimal policy parameters. We experiment with the different combinations of the following values of σ, k and k_0:

- $\sigma = 4, 8, 12$
- $k = 200, 400, 1000$
- $k_0 = 0, 5, 10, 20, 30, 50$

We test each combination of these values on 100 randomly generated sample paths and collect the data from each run. We then report the average cost from the 100 sample paths. In addition, for several sample paths, we compare our results with the k-period integer program, SSC-DL($\bar{X}_k$), as a benchmark. Note that due to the time it takes to solve SSC-DL($\bar{X}_k$) (on the order of days), it is not practical to perform this test for all 100 sample paths. We use the Kiefer-Wolfowitz updating method outline in Section 3.1 with initial gradient estimation and search step sizes $a = 2$ and $c = 2$. We also use the termination criteria outlined in Section 3.1 for each of our cases. Below, we report on our observations.

Results and Observations

Impact of Sample Path Length

We begin by exploring the performance of the math program SSC-DL($\bar{X}_k$) as the horizon length increases (so in this section, we do not use stochastic gradient search). We know that this MILP converges to the optimal solution (in this case, as derived in the recursive formulation above) as the length of the sample path goes to infinity, but we curious how the formulation performs for finite sample path lengths. As it takes days to solve each instance of SSC-DL($\bar{X}_k$) for long horizon lengths, for this set of experiments, we focus on a single set of demand distribution parameters – $Normal(32, 8)$ truncated to non-negative values – and observe the solution obtained by SSC-DL($\bar{X}_k$) over seven different sample paths as the length of these paths increase in size. We compare the per-period cost (that is, the objective function) of SSC-DL($\bar{X}_k$) for various values of k to the optimal solution, in this case, the per-period cost of 290. The expected cost per period is very close over the different sample path lengths, but the variance reduces as the sample path gets longer. Table 3.1 shows the mean and standard deviation of the cost-per-period objective function for different retrospective sample path length, over 7 independently generated demand sample paths.

Table 3.1: Cost Per Period, Normal(32, 8)

Sample Path Length	Mean	Std Dev
200-Period	288	9.54
400-Period	289	6.86
1000-period	290	4.48

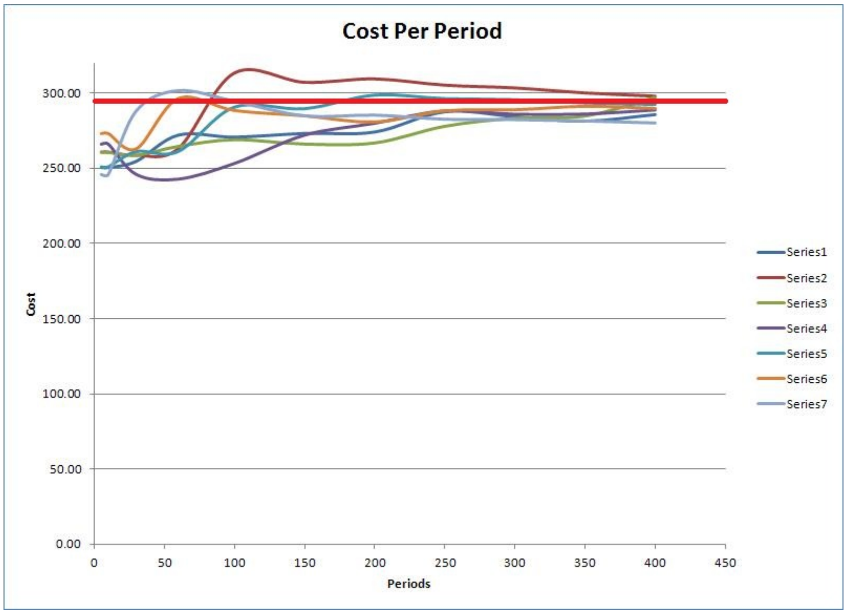


Figure 3.2: Cost Per Period, Normal(32, 8)

In Table 3.1 and Figure 3.2, we see that as the sample path length k increases, the cost per period determined by the retrospective model, SSC-DL($\bar{X}_k$), over the different sample paths, $\bar{X}_k$, approaches the true expected cost and decreases in variance. This gives some sense of the length of sample path necessary to get a solution close to the actual optimal solution from the Clark and Scarf [7] model.

Speed of the ROGS Approach

The solution to SSC-DL($\bar{X}_k$) converges to the steady state optimal solution of the recursive formulation outlined in section 4.1.2 as k increases in size. However, SSC-DL($\bar{X}_k$) with large k takes days for solvers such as CPLEX to optimize. We demonstrate the speed of the ROGS algorithm by comparing the solution times required and the optimal solutions obtained from the steady state recursive formulation, the solution of a large SSC-DL($\bar{X}_{400}$), the solution of a short SSC-DL($\bar{X}_{30}$), and the result of ROGS. Due to the solution time required to solve SSC-DL($\bar{X}_{400}$) , we only experiment with a $Normal(32, 4)$ demand distribution. Experimental results suggest that ROGS obtains a solution very close to that of the optimal solution (which is found using the recursive formulation, as well as by the large MILP), and does so very efficiently.

Specifically, Table 3.2 compares the performance on a single sample path generated from a $Normal(32, 4)$ distribution, of the optimal solution (found analytically), the

400-period IP solution, the truncated 30-period IP solution and the gradient search result using the 30-period IP solution as the initial point.

Table 3.2: Gradient Search : Init = 30 period, horizon = 400 period

Method	S1	S2	S3	S4	Cost Per Period	Time	%Error
Steady State Optimal	40	34	34	32	241.76	–	–
SSC-DL($\bar{X}_{400}$)	40.350	37.791	30.752	31.286	241.76	33.5 hr	0%
SSC-DL($\bar{X}_{30}$)	39.121	33.378	29.339	33.976	259.15	17 sec	7.191%
ROGS Algorithm	41.121	36.168	30.339	33.405	242.72	12 sec	0.397%

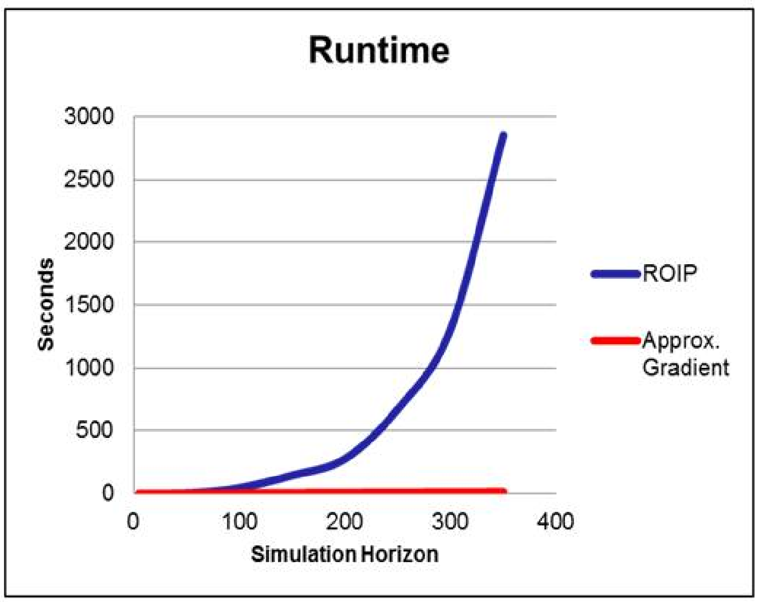


Figure 3.3: Gradient Search : Init = 30 period, horizon = 400 period

Using CPLEX, the time to solve SSC-DL($\bar{X}_k$) gets quite large as k increases. With $k = 400$, it takes our machine 33.5 hours to optimize SSC-DL($\bar{X}_{400}$) with CPLEX. Using ROGS, we are able to reduce the computational times significantly while keeping the accuracy within 0.4%.

Starting Point for Gradient Search

ROGS utilizes a gradient descent method combined with MILP. Here, we explore whether gradient search alone yields a poor solution when starting from an arbitrary initial solution, and if increasing the initial gradient search step sizes can compensate for a bad initial starting point.

For a given distribution, we simulate 100 sample paths and observe the average performance, including the optimal cost per period obtained and the number of gradient search iterations, using an initial solution obtained by an k_0-period truncated ROIP. Extracted from Appendix A, Tables 3.3 and 3.4 present the results of two experiments using $Normal(32, 8)$ and $Normal(32, 12)$ distributions, respectively, truncated and normalized (so that the probability density sums to 1) to non-negative numbers for demand. The examples utilize a retrospective sample path length of 400 periods. The gradient search is also limited to 2000 iterations in each case. For the two-step algorithm, we use an initial gradient estimation and search step size $c_0 = 2$. For comparison, we also test the case using a trivial zero-basestock starting point with initial gradient search step size $c_0 = 10$.

Table 3.3: Normal(32, 8)

k_0	0	5	10	20	30	50	Steady State	$c_0 = 10$
SSC-DL($\bar{X}_{30}$) Solution	4762	498	353	319	308	299	-	-
ROGS Solution	4203	342	297	294	292	292	291	1841
% Error from Steady State	1343%	17.3%	2.0 %	0.8%	0.3 %	0.2%	0.0%	626%
# Iterations	1714	1158	437	134	56	42	-	-
% Forced Termination	0%	46%	10%	1%	0%	0%	-	-

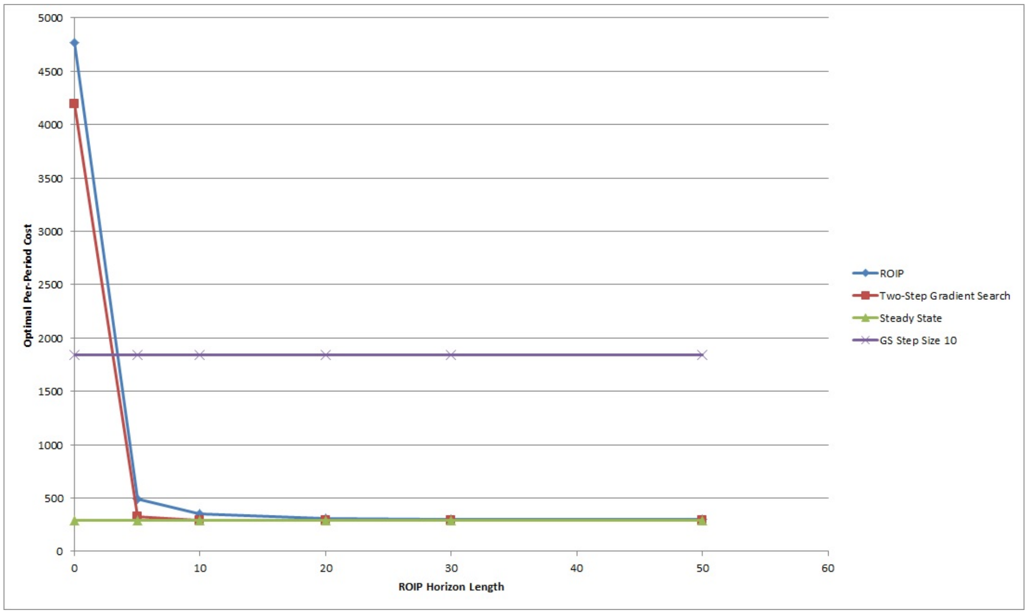


Figure 3.4: Normal(32, 8)

Table 3.4: Normal(32, 12)

k_0	0	5	10	20	30	50	Steady State	$c_0 = 10$
SSC-DL$_{short}$ Solution	4786	625	430	377	366	348	-	-
Gradient Search Solution	4210	446	356	343	340	339	337	1873
% Error from Steady State	1151%	32.5%	5.8 %	2.0%	1.0 %	0.7%	0%	556%
# Iterations	1706	1319	775	409	245	99	-	-
% Forced Termination	0%	52%	17%	4%	1%	0%	-	-

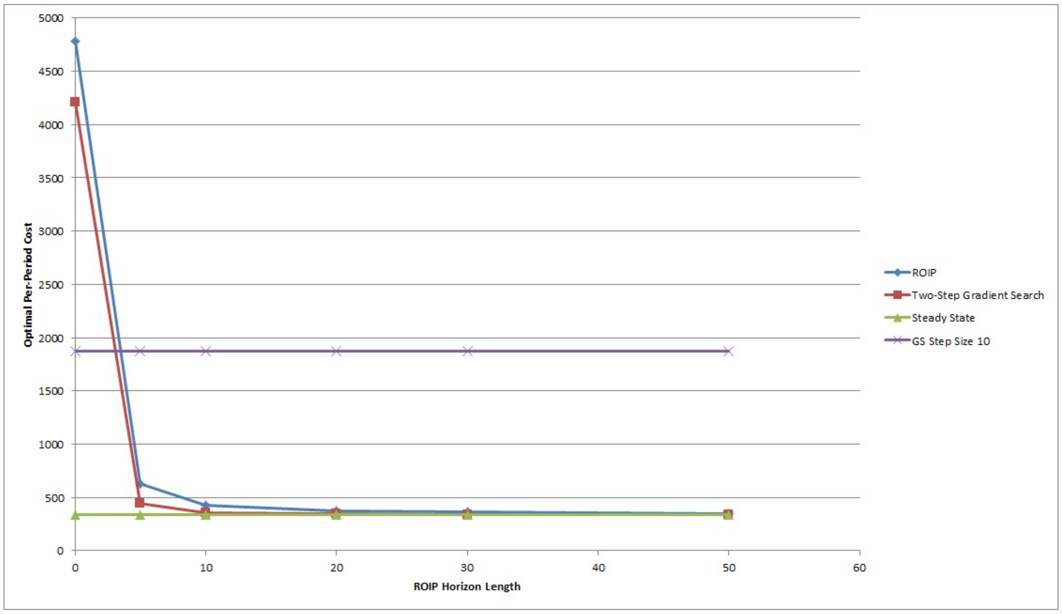


Figure 3.5: Normal(32, 12)

Here, we make two important observations from Tables 3.3 and 3.4 First, a larger k_0 reduces the gap between the solution obtained by ROGS and the reference solution obtained by the recursive stochastic formulation. However, a larger k_0 also requires a longer computation time for the first step of ROGS. Also, it is important to start the second step of ROGS with a good starting point solution. With a trivial solution, ROGS is unable to obtain a good solution even with a much larger gradient estimation and search stepsize.

Alternative Approaches to Starting Point

While using a trivial starting point for the gradient search is shown to be ineffective, we suspect that a random sample approach may be effective as well. To demonstrate this, we randomly generate 100 combinations of the 4 basestock levels to use as the starting point for gradient search. Each basestock level is limited to a random number between 0 and 100 following a uniform distribution. We evaluate the average cost per period using a 400-period demand sample path. Table 3.5 shows the best solution obtained by this approach in contrast to the average solution obtained by the ROGS approach.

Table 3.5: Normal(32, 12): Random Starting Point

k	Cost Per Period	# Iterations	Run Time
ROGS (Average)	339	123	56 sec
Random Starting Point (Best)	332	789	2.5 hr

With only 4 decision variables, this random approach is able to provide a good random starting point for the gradient search chosen from 100 random samples, however, it requires significantly longer computational time when compared to using the starting point obtained by a short retrospective MILP. Additional computation time is required by both evaluating the total cost of the randomly generated decision variable values and the additional gradient search iterations. The results suggest that in our setting, it is more efficient to use the short retrospective MILP to obtain the starting point for gradient search, while this random approach may be used as an alternative when the MILP becomes more complex or when the model becomes non-linear. This random approach will also be likely to require more randomly generated samples as the number of decision variables increase thus requiring even longer computation times.

Impact of Variance in Demand

As the stochastic parameters in a system increase in variance, the optimal solution becomes more difficult to find. In this set of experiments, we explore how variance in demand impacts the performance of ROGS. We use $Normal(32, 4)$, $Normal(32, 8)$ and $Normal(32, 12)$ demand with different values of k_0. We then compare the 100 run average optimal solution obtained by ROGS in each case, and record the percentage gap with the solution obtained from the Clark and Scarf (1960) [7] model.

In our experiment, we observe that the gradient search requires better initial solutions for higher variance distributions. In general, the initial value provided by the 20 or 30-period ROIP enables the gradient search to find a good solution in all cases, but for low variance distributions, a shorter run is sufficient. Extracted from Appendix A, Table 3.6 demonstrate how quickly the gap closes for three normally distributed demand sample path with standard deviations 4, 8, and 12.

Table 3.6: Gradient Search Gap

k_0:	5	10	20	30	50
$\sigma = 4$	0.98%	0.27%	0.26%	0.25%	0.27%
$\sigma = 8$	17.33 %	2.02%	0.79%	0.26%	0.22%
$\sigma = 12$	32.47 %	5.81%	1.98%	1.02%	0.70%

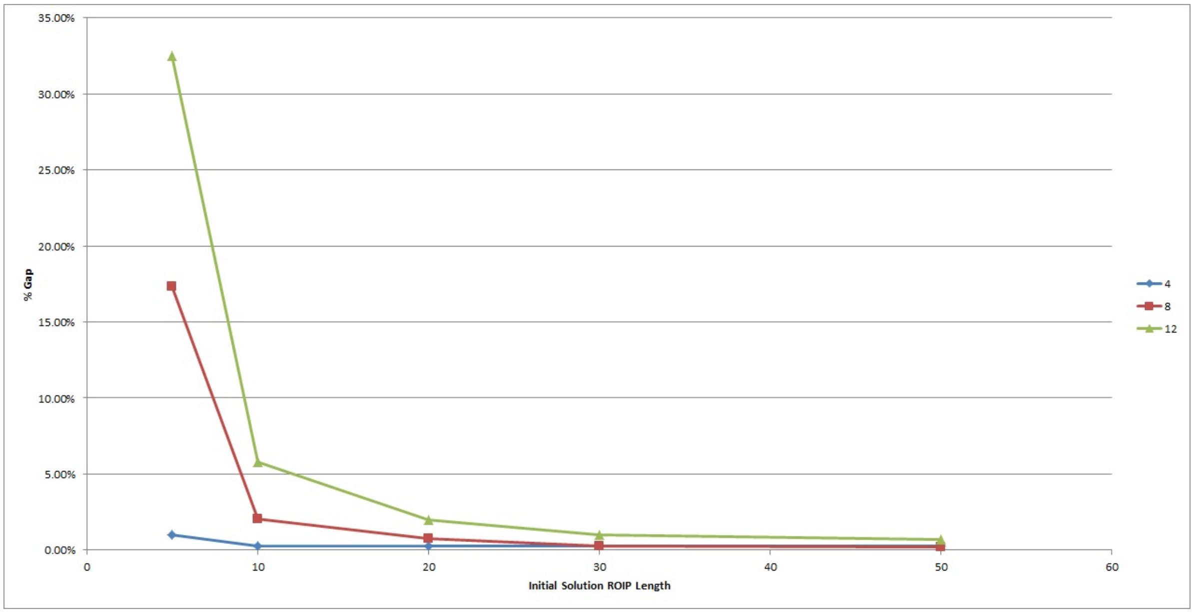


Figure 3.6: Gradient Search Gap

We observe that in order for ROGS to obtain a good solution, a larger k_0 is required as the variance of the random demand increases. We also conjecture from this observation, that a larger k_0 will also be needed when we consider models with additional sources of uncertainty such as lead times between stages.

Serial Supply Chain with Stochastic Lead Time

Problem Statement

In the previous section, we presented a set of experiments exploring the performance of ROGS for a problem with a known optimal solution. We now extend our application to serial supply chain systems with stochastic lead times, Model SSC-SL($\bar{X}_k$), which is much more difficult to optimize.

We again use a 4-stage serial supply chain as our test case. The parameters and decision variables are similar to the previous section. To model the random lead times, we introduce a lead time matrix L^i for each stage, where $L^i_{s,t}$ is equal to 1 if and only if the shipment leaving stage i at time s arrives at stage $i-1$ at time t, and 0 otherwise. This replaces the variable L^i_t in SSC-DL($\bar{X}_k$) and will be included in the generated variables, $\bar{X}_k$. We carry over the parameters and variables from SSC-DL($\bar{X}_k$) and define the following additional parameter for random lead times:

- L_t^i = lead time matrix for outgoing shipments of stage i

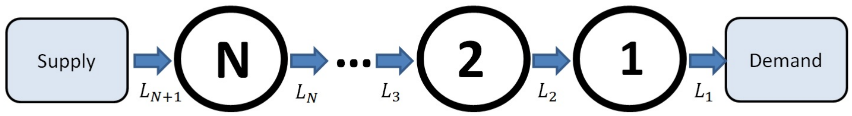


Figure 3.7: SSC-SL

The mixed integer program, SSC-SL, has the same objective function as SSC-DL (3.1), as many of hte same constraints. Constraints (3.2) and (3.4) from SSC-DL are replaced as described below:

SSC-SL($\bar{X}_k$):

$$\begin{aligned} min \quad & (3.1) \\ s.t. \quad & T_t^i = \min\{I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1}, d_t + B_{t-1}^i\} \quad i = 1, ..., N \;, \quad t = 1, ..., k \qquad (3.12) \\ & I_t^i = I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1} - T_t^i \quad i = 1, ..., N \;, \quad t = 1, ..., k \qquad (3.13) \\ & (3.3), (3.5) - (3.11) \end{aligned}$$

As before, we introduce integer variables and a big constant M to model the minimization constraint (3.12) above:

$$\begin{aligned} I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1} &\leq d_t + B_{t-1}^i + zM \\ d_t + B_{t-1}^i &\leq I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1} + (1-z)M \\ T_t^i &\leq I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1} \\ T_t^i &\leq d_t + B_{t-1}^i \\ T_t^i &\geq I_{t-1}^i + \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1} - zM \\ T_t^i &\geq d_t + B_{t-1}^i - (1-z)M \end{aligned}$$

Experimental Design

In contrast to SSC-DL($\bar{X}_k$), supply chain systems with stochastic lead times between stages are complex and difficult to optimize. Thus, we evaluate the performance of our algorithm by comparing it to an easy-to-find feasible solution. We obtain a feasible solution by solving a deterministic lead-time version of the model where we let the lead time always equal its minimum. We then use the resulting basestock levels as the operating parameters in the SSC-SL($\bar{X}_k$) model. For this set of experiments, we again use periodic demand with normal distributions. We focus our analysis on the different lead time distributions rather than the different demand distributions. We let the i.i.d. periodic demand distribution be normally distributed with mean 32 and standard deviation 4, and let the lead time between stages follow a discretized, truncated normal distribution with mean 2. We experiment with lead time standard deviations 0.3, 0.6 and 1.0. We use unit incremental holding costs ($h_1 = 4, h_2 = 3, h_3 = 2, h_4 = 1$), and a backorder penalty cost of 36 per unit ($\pi = 36$) as before.

Due to the time it takes to optimize SSC-SL($\bar{X}_k$) with CPLEX, we were not able to optimize SSC-SL with $k > 30$. We use $k_0 = 30$ for the first step of ROGS, and experiment with different values of $k = 50, 100, 200, 400, 1000$ for the second step of our algorithm using the Kiefer-Wolfowitz updating method with initial values of $a = 30$ and $c = 30$. We again let δ be 0.0001% of the initial objective function value and limit the number of iterations to 2 hours of computational time. We then use Monte Carlo simulation to evaluate the performance of our results over 100 randomly generated sample paths. We report our observations from these experiments in the next section.

Results

Comparison with Simple Heuristic Solution

We obtain a set of basestock levels as a heuristic by formulating an instance of SSC-DL with the lead times always equal to 1 (the minimum lead time). We then use those basestock levels as the operating parameter in the SSC-SL($\bar{X}_k$) model. Due to the required computational times to optimize SSC-SL($\bar{X}_k$), we repeat the experiment for each value of standard deviation only ten times. In each instance, we evaluate our solution using Monte Carlo simulation of 100 randomly generated sample paths. Tables 3.7 through 3.9 show the average optimal solution value obtained by our two-step approach over the 10 experiments.

Table 3.7: Stochastic Lead Times, $\sigma = 0.3$

Lead Time	S_1	S_2	S_3	S_4	Cost Per Period
Heuristic Solution	40	34	34	32	2595
SSC-SL($\bar{X}_{30}$)	32	26	32	27	3196
ROGS	78	49	32	32	1046

Table 3.8: Stochastic Lead Times, $\sigma = 0.6$

Lead Time	S_1	S_2	S_3	S_4	Cost Per Period
Heuristic Solution	40	34	34	32	4318
SSC-SL($\bar{X}_{30}$)	35	32	32	31	4701
ROGS	58	68	51	31	2339

Table 3.9: Stochastic Lead Times, $\sigma = 1.0$

Lead Time	S_1	S_2	S_3	S_4	Cost Per Period
Feasible Solution	40	34	34	32	6916
SSC-SL($\bar{X}_{30}$)	33	31	31	21	7730
ROGS	49	84	31	21	5497

We see from Tables 3.7, 3.8 and 3.9 that in all cases, the short ROIP, SSC-SL($\bar{X}_k$), alone does not provide a better solution than the heuristic solution. However, ROGS is able to improve the solution through gradient search. We also see that the average performance of ROGS is better the feasible solution. The end result shows an optimal solution that favors increasing inventory level at the most downstream stage as the stochastic lead time increases in variance. This is expected because random lead times make it more difficult to depend on upstream inventory.

As the lead time variance increases, the optimal basestock levels also become harder to find. In the case where $\sigma = 0.3$, ROGS is able to provide a solution that on average costs 60% less than the solution obtained by the heuristic. When $\sigma = 1.0$, the average solution found by ROGS only costs 21% less than the heuristic solution. In addition, observe that even the deterministic lead time problem is challenging to solve with long lead times due to the longer tail of the distribution – for example, if we attempt to solve the recursive formulation for serial supply chain with $L = 3$, the solution time will require over a year of computation.

Size of Sample Path

Recall that we are unable to increase k_0 for SSC-SL($\bar{X}_{k_0}$) in the first step of ROGS because the problem cannot be solved in a reasonable amount of time. We are, however, able to increase k for the large SSC-SL($\bar{X}_k$) formulation used in the second step (gradient search) of ROGS. We experiment with sample paths of different sizes used in the gradient search, $k = 50, 100, 400, 1000$, and observe the effects on the solution. Table 3.10 show the per-period cost obtained by ROGS with different values of k:

Table 3.10: Effects of k

Lead Time	S_1	S_2	S_3	S_4	Cost Per Period
ROIP 30 Periods	40	56	32	34	3675
ROGS on SSC-SL($\bar{X}_{50}$)	56	56	32	34	3123
ROGS on SSC-SL($\bar{X}_{100}$)	56	56	32	34	3123
ROGS on SSC-SL($\bar{X}_{400}$)	56	56	32	34	3123
ROGS on SSC-SL($\bar{X}_{1000}$)	56	56	32	34	3123

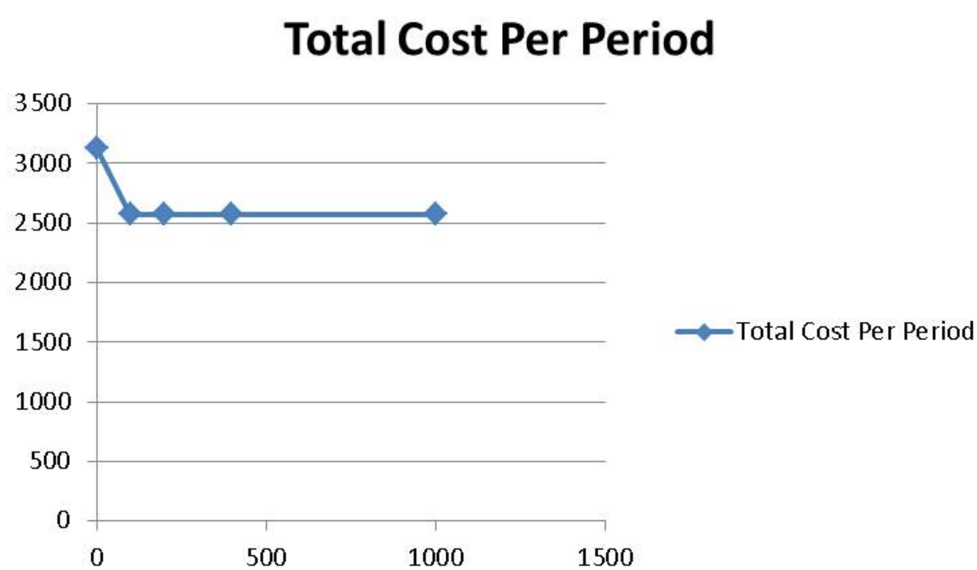


Figure 3.8: Effects of k

In contrast to the effect of increased k_0 in SSC-SL($\bar{X}_{k_0}$), the benefit of a larger k in SSC-SL($\bar{X}_k$) is less apparent. The gradient search algorithm yields the same results from a 50 period retrospective sample path, a 1000 period retrospective sample path, and everything in between. We conjecture that one of two things is happening here:

either the global optimum converges after 50 periods, or the gradient search step is trapped in a local optimum.

Starting Point of Gradient Search

To demonstrate the importance of the first step of ROGS, we examine the solutions resulting from the solution of a 30-period ROIP and a trivial solution, zero basestock levels at all stages, as the starting point of our gradient search step. Note that the solution time required to optimize SSC-SL($\bar{X}_k$) is much longer than that of SSC-DL($\bar{X}_k$), hence we are only able to use up to $k_0 = 30$. Luckily, the solution speed of the second step of ROGS for SSC-SL($\bar{X}_k$) is very similar to that for SSC-DL($\bar{X}_k$), as the only added complexity involves generating the sample paths. Obtaining the total cost at each step remains computationally inexpensive.

Table 3.11 shows the results, both with the starting point obtained by solving SSC-SL($\bar{X}_{30}$) and a trivial starting point of basestock levels all equal to 0:

Table 3.11: Effects of Starting Point

Lead Time	S_1	S_2	S_3	S_4	Cost Per Period
ROGS with SSC-SL($\bar{X}_{30}$)	52	32	32	31	3344
Search with Trivial Starting Point	27	0	0	0	10658

By using the trivial starting point, the gradient descent method obtains a solution of 10658 as opposed to the far better solution of 3344 obtained by ROGS. This suggests that using the solution to a short SSC-SL($\bar{X}_{30}$) as a good starting point for the gradient search algorithm is critical.

Effects of Gradient Search Step Size

With longer lead times, it is natural to assume that the optimal solution has higher basestock levels. With this in mind, we experiment with initial step sizes used in the second step (gradient search) of ROGS. In this experiment, the lead times between the 4 stages are normally distributed with mean 2 and standard deviation 1, and are discretized and truncated at 0 and 4. Table 3.12 shows the optimal solutions obtained from different initial gradient search step sizes.

Table 3.12: Effects of Gradient Estimation Step Sizes

Initial Step Size	S_1	S_2	S_3	S_4	Cost Per Period
SSC-SL($\bar{X}_{30}$)	31	31	31	27	7278
$C_0 = 3.5$	59	31	31	27	6134
$C_0 = 5.0$	38	65	31	27	5984
$C_0 = 15$	92	91	32	27	2971
$C_0 = 25$	153	73	31	27	1751
$C_0 = 30$	125	95	83	61	534
$C_0 = 50$	124	100	89	74	498
$C_0 = 100$	131	81	89	78	504
$C_0 = 200$	231	31	31	127	554
$C_0 = 300$	331	31	31	27	610

As the initial gradient estimation step size, C_0, increases, ROGS is able to obtain a better optimal solution for SSC-SL($\bar{X}_{1000}$). However, after a point, as C_0 increases, the solution obtained becomes worse. There appears to be a "sweet spot" for initial gradient search stepsize for each problem that depends on its parameters.

Step Size vs. Starting Point

We saw in the section 3.2 that a properly chosen step size can result in better solutions. We further examine the tradeoff between larger gradient search stepsizes and a good starting point of the search obtained by step one of ROGS. We test ROGS with leadtime between stages generated by a discretized, truncated normal distribution with mean 2 and standard deviation 1. We use various initial step sizes between 3 and 30. We then compare the performance of using a trivial starting point (zero basestock levels) and a starting point obtained by optimizing the short SSC-SL($\bar{X}_{30}$). Table 3.13 and figure 3.9 show the results

Table 3.13: Tradeoff between Step Size and Starting Point

Initial Step Size	SSC-SL($\bar{X}_{30}$)	Zero-Basestock
3.5	6134	10658
5	5984	10195
15	2971	7177
25	1751	4369
30	534	2924
100	504	520
200	554	597
300	610	706

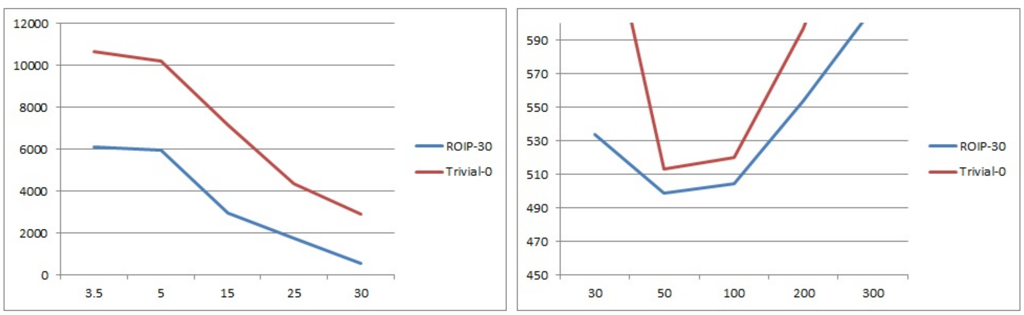


Figure 3.9: Tradeoff between Step Size and Starting Point

While a larger step size certainly helps the gradient search find a better solution, it finds an even better solution with a good starting point. This further highlights the importance of the first step of our two-step algorithm.

Concluding Remarks

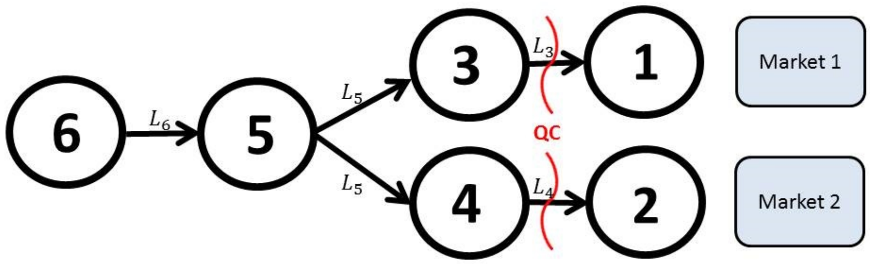
We successfully applied our two-step algorithm to a serial supply chain system that is challenging to solve. We confirmed the importance of both of the steps in our two-step ROGS algorithm. Furthermore, we highlighted the importance of choosing a good gradient step search size. In addition to the experiments described above, we also experimented with some modification to the Kiefer-Wolfowitz algorithm. For instance, we conditionally decreased the gradient estimation and incremental step size only when changing the direction of improvement; However, initial results are very similar to simply starting with a larger initial step sizes, c and a. In the next section, we apply ROGS to more complex supply chains.

3.3 Complex Supply Chain

In the previous section, we demonstrated the performance of our two-step ROGS algorithm on serial supply chain models. Our research was originally motivated by the desire to understand inventory positioning in the daily operation of supply chains when our ultimate goal is to be able to implement strategies to deal with supply chain disruptions. A firm might, for instance, have additional safety stock levels at a subset of production steps to deal with an unlikely event that disrupts parts of the supply chain. Such disruptions may be caused by quality control failures, regulation changes, natural disasters, human errors, and more. The question we were (and are) interested in addressing is how to make normal day-to-day operating decisions in supply chains when our goal is to both minimize daily operating costs, and ensure that customers are not impacted when some sort of disruption arises. In this section, we demonstrate the flexibility and applicability of our algorithm by using it to optimize a highly complex biopharmaceutical supply chain under a basestock policy. The model involves the steps of fermentation, purification, distribution, labeling for two separate markets, quality control inspection, and delivery to the end retailer. We will present the model formulation as well as the optimal solution obtained by ROGS.

Problem Statement

We consider a hypothetical biopharmaceutical manufacturing operation that delivers one drug to two markets. The drug needs to go through fermentation and purification (stage 4). Then they will need to be filled into vials (stage 3) for labeling (stage 2). Each market requires its own label and quality control inspection (e.g. government regulatory inspection) before each batch of the drugs can be distributed to customers (stage 1). We assume the drug will pass quality control with a fairly high probability, α. However, when it fails, all the drugs of that batch needs to be discarded. This potential disruption caused by quality control will cause the optimal inventory level to be higher to avoid shortages. The tradeoff between overstocking and the risk of stockout is what we will be focusing on in this section.



Biopharmaceutical Supply Chain

For this test, we let the demand in both markets be normally distributed with mean 32 and standard deviation 4. We use incremental holding costs that increase downstream. Specifically, we let the holding cost be 1 at stage 6, 2 at stage 5, 3 at stages 3 and 4, and 4 at stage 1 and 2. We let the lead time between stages be normally distributed with mean 2 and standard deviation 1. We assume $\alpha = 0.95$ (i.e. the drugs will pass quality control and reach their corresponding destination market with probability 0.95). Furthermore, we assume that no two batches can fail within seven periods of each other. We make the second assumption for two reasons. The first is that the first step of our algorithm requires a "long enough" ROIP that captures the steady state of a system. The rare occurrence of two consecutive QC failures is most likely not going to be captured with an ROIP unless the retrospective horizon is huge. The second is that in practice, the firms with which we have worked do not consider multiple supply chain disruptions within a short period of time because it is operationally infeasible to keep such high inventory.

Model

We use previously defined notation, and define the following additional parameters for this model:

- d_t^j = demand in period $t = 1, 2, ..., k$ for market $j = 1, 2$
- $\alpha_t^i = 1$ if QC passes for the batch coming out of stage 2 for market i at time t, 0 otherwise

and the following additional variables for modeling purposes:

- w_t^i = the order amount (wanted) at time t at stage i
- T_t^i = inventory shipped at time t from stage i
- R_t^i = inventory received at time t at stage i

The math program is then formulated as follows:

BSC($\bar{X}_k$):

$$min \quad \widehat{f} = \frac{1}{k}\Big[\sum_{t=1}^{k}\pi(B_t^1+B_t^2)+\sum_{t=1}^{k}\sum_{i=0}^{6}h^i I_t^i\Big]$$

$$s.t. \quad R_t^i = \sum_{s=1}^{t-1} L_{s,t}^{i+1} T_s^{i+1}, \qquad i=5,6\ ;\ \ t=1,...,k \tag{3.14}$$

$$\sum_{i=3,4} R_t^i = \sum_{s=1}^{t-1} L_{s,t}^{5} T_s^{5}, \qquad t=1,...,k \tag{3.15}$$

$$R_t^i = \sum_{s=1}^{t-1} \alpha_s^i L_{s,t}^{i+2} T_s^{i+2}, \qquad i=1,2\ ,\ \ t=1,...,k \tag{3.16}$$

$$w_t^i = d_t^i + \sum_{s=1}^{t-1}(1-\alpha_s^i) L_{s,t}^{i+2} T_s^{i+2}, \qquad i=1,2\ ,\ \ t=1,...,k \tag{3.17}$$

$$T_t^i = min\{I_{t-1}^i + R_t^i, \sum_{j=1,2} d_t^j + B_{t-1}^i\} \qquad i=5,6\ ;\ \ t=1,...,k \tag{3.18}$$

$$T_t^4 = min\{I_{t-1}^4 + R_t^4, w_t^2 + B_{t-1}^i\} \qquad t=1,...,k \tag{3.19}$$

$$T_t^3 = min\{I_{t-1}^3 + R_t^3, w_t^1 + B_{t-1}^i\} \qquad t=1,...,k \tag{3.20}$$

$$T_t^2 = min\{I_{t-1}^2 + R_t^2, d_t^2 + B_{t-1}^i\} \qquad t=1,...,k \tag{3.21}$$

$$T_t^1 = min\{I_{t-1}^1 + R_t^1, d_t^1 + B_{t-1}^i\} \qquad t=1,...,k \tag{3.22}$$

$$B_t^i = \sum_{j=1,2} d_t^j + B_{t-1}^i - T_t^i \qquad i=5,6\ ,\ \ t=1,...,k \tag{3.23}$$

$$B_t^4 = w_t^2 + B_{t-1}^4 - T_t^4 \qquad t=1,...,k \tag{3.24}$$

$$B_t^3 = w_t^1 + B_{t-1}^3 - T_t^3 \qquad t=1,...,k \tag{3.25}$$

$$B_t^2 = d_t^2 + B_{t-1}^2 - T_t^2 \qquad t=1,...,k \tag{3.26}$$

$$B_t^1 = d_t^1 + B_{t-1}^1 - T_t^1 \qquad t=1,...,k \tag{3.27}$$

$$I_t^i = I_{t-1}^i + R_t^i - T_t^i \qquad i=1,...,6\ ,\ \ t=1,...,k \tag{3.28}$$

$$T_t^7 = \sum_{j=1,2} d_t^j \qquad t=1,...,k \tag{3.29}$$

$$(3.6)-(3.11)$$

The objective function seeks to minimize the sum of all holding costs plus the total backorder penalty incurred in both market 1 and market 2. Constraint (3.14) defines the amount of inventory received in stages 5 and 6, similar to a serial supply chain model. Constraint (3.15) ensures the flow balance from stage 5 to stage 3 and

4. Constraint (3.16) defines the amount of inventory received in stages 1 and 2 (for market 1 and 2 respectively) according to the quality control pass rate α.

Constraint (3.17) calculates the desired order quantity for stages 1 and 2, effectively reordering the batches that fail QC. Constraints (3.18) through (3.22) calculate the amount shipped from each stage at each time period. Constraints (3.23) through (3.27) calculate the amount of backorder incurred at each stage, where some are concerned with the aggregate demand of both markets some are only concerned with the demand of one of the markets. Constraint (3.28) updates the amount of inventory in each stage at each time period. And finally, constraint (3.29) creates a dummy external supplier that always satisfies demand.

As in previous models, we replace each minimization constraint with a set of inequalities by introducing integer variables and a large constant M.

Computational Result and Discussion

To optimize this biopharmaceutical supply chain model over 1000 periods, we employ ROGS to solve BSC($\bar{X}_{1000}$). As a benchmark, we first evaluate the average cost per period using a set of trivial basestock levels, namely the average demand of 3 periods across all stages. BSC($\bar{X}_{k_0}$) is significantly more complex than the other models we consider in this paper, so CPLEX is only able to solve for up to $k = 20$ (any larger values of k_0 results in our computer running out of memory). Therefore, for our ROGS algorithm, we formulate a short version, BSC($\bar{X}_{20}$) and optimize it using CPLEX. Then, we use the solution obtained from CPLEX as a starting point to perform gradient search to optimize BSC($\bar{X}_{1000}$). In this experiment, we let $a = 30$ and $c = 30$. We continue to let the termination threshold, δ, be 0.0001% of the initial objective function value and limit the number of iterations to 2 hours of computational time. With 10 i.i.d. simulated sample paths, our algorithm returns the following basestock levels:

Table 3.14: Biopharmaceutical Supply Chain

Solution Method	s_1	s_2	s_3	s_4	s_5	s_6	Cost Per Period
Benchmark	96	96	96	96	192	192	2281
ROGS	87	140	137	42	71	40	1694

In this test case (shown in Table 3.14), ROGS is able to provide a 26% improvement in average cost per period, over the benchmark. However, through our experiments, we had to manually filter out the "bad" sample paths used in the first step of ROGS, namely the ones that experience no QC disruption in the first 20 periods. These sample paths provide an overly optimistic scenario leading to a low perceived optimal basestock level as the starting point of the gradient search. In the long sample path used in the second step of ROGS, however, the disruptions are generated with no filtering. It is important to emphasize that ROGS only works if the short sample path used in the first step sufficiently captures the steady state behavior of the system in the long run. As the probability of supply chain disruption increases, a larger k_0 will be required, and the "good" sample paths will require a stricter filter.

3.4 Conclusions

Our two-step ROGS solution approach was motivated by the need to quickly and efficiently optimize supply chain operation parameters in a wide variety of complex, stochastic supply chain layouts. By combining retrospective optimization and stochastic gradient search, we have developed a flexible algorithm, ROGS, that is able to model the steady state operation of such supply chains. ROGS can then obtain near-optimal operating parameters for these models efficiently and effectively. Our experiments demonstrate the flexibility and performance of ROGS, and give insight into the process of fine tuning the ROGS algorithm. In particular, we are able to find good solutions to complex supply chain models that have previously been too complex to analyze effectively using standard commercial solvers. In the next chapter, we explore the flexibility of the ROGS algorithm by applying it to a data-driven approach for supply chain optimization. Furthermore, we extend the model complexity by including a state-dependent inventory policy and modeling perishable inventory.

Chapter 4

Data-Driven Perishable Inventory Optimization

Supply chain optimization approaches discussed up to now assume a known underlying probability distribution for any random variables. This allows the Retrospective Optimization approach to use a generated sample path to optimize the decision variables. However, for supply chain problems where the underlying distributions of the random variables are unknown to the decision maker, a data-driven optimization approach can be used. In recent years, researchers increasingly use a data-driven approach [13] [18] [22] to optimize supply chain operations. Similar to Retrospective Optimization, this approach models the supply chain with a set of *realized* random variables. Different from Retrospective Optimization, however, past data are used in place of randomly generated values. Supply chain systems can be modeled as deterministic MILP's with this data-driven approach and these MILP's present a similar challenge as we saw earlier in modeling supply chains in the biopharmaceutical industry. Therefore, we can apply the previous ROGS solution approach to overcome the challenges of these data-driven models.

We test our solution approach using data from a major European grocery chain store. Since the data comes from a system that deals with perishable inventory, we build our models to include expiration. Perishable inventory provides a variety of challenges that increases the complexity of the problem. For example, it is important to distinguish between inventory with different ages. It is also important to enforce an implementable consumption rule, such as first-in-first-out (FIFO). Without such rules, using historical data, the model may decide to "cheat" by selling newer inventory and discarding older inventory to avoid unnecessary holding costs several periods into the future. Furthermore, for supply chain systems where a single warehouse provides supply for a large number of retailers, the distribution policy of limited inventory between these retailers is often difficult to model as an MILP and almost impossible to solve. Fortunately, our ROGS algorithm is designed to be flexible and adaptable to different structures of supply chains under a wide variety of assumptions. In this chapter, we demonstrate this flexibility by applying our algorithm to obtain effective inventory policy settings for a grocery retail chain operation.

4.1 Problem Setting

Inventory management for perishable products such as fruit and vegetables is a challenging task for retail managers. If a retailer orders too many items in advance, the retail company incurs inventory holding costs and eventually additional costs for discarding expired products. If the retail manager orders too few items, the customers do not find the products they were looking for in the store and the retail company loses potential sales. The retailer's goal is to determine the optimal order quantity that balances inventory holding and stockout costs.

If a product has a shelf-life of several periods, the retailer should consider how many items are in-stock when placing a new order. The current stock can be characterized by how many items are still available and when each one of these items is going to perish. Not necessarily all items of one age perish at the same point in time. It is common in practice that some items of one kind already perish after one day whereas others that arrived at the same point in time last for several days. There are many reasons for this; for example, customers or store employees not handling products with care, quality discrepancy or differing modes of transportation. As a result, only a fraction of inventory available at the end of one day is carried over to the next day and the rest has to be discarded.

Another important aspect of the ordering decision is whether the stock should be held at the retail stores or the central warehouse. Often in a retail setting, the different types of products are delivered from many individual suppliers to a central warehouse. The products are then sorted, consolidated and distributed according to individual store orders which are loaded on trucks that deliver them to the stores. The retailer can choose between at least two possible locations where stock can be held: either at the warehouse or at the retail stores. On the one hand, holding stock at the warehouse allows the balancing of fluctuations in demand between stores. Additionally, warehouses are often equipped with compartments of different temperatures, humidity levels, etc. that accommodate the different requirements of fresh products. As a result, the shelf life of fresh products may be longer if they are held at the warehouse as opposed to the shelves in a store. On the other hand, demand is stochastic and a store might stockout before the new replenishment arrives if the product is delivered from the warehouse as opposed to holding stock at the store itself.

One approach for determining optimal order quantities is to first estimate the underlying demand distribution for each product and then determine the optimal inventory level. This approach requires several assumptions, such as a demand distribution which needs to be fitted to each dataset and may differ between stores or products. A data-driven approach, however, can integrate demand estimation and inventory optimization. This approach eliminate the need to fit a distribution for the demand. We determine the inventory policy based on historical sales data by balancing the retail manager's trade-off between ordering too many and too few. The optimal inventory policy depends on the shelf-life of a product, costs and/or service level requirements. This approach is not only applicable to food products like fruit and vegetables, but also to blood or pharmaceuticals, where perishability is a crucial factor.

It is important to note that, using this data-driven approach to optimize supply chain operations typically results in MILP models with a large number of integer variables and constraints. Because of this, most related work has considered only single stage settings. In order to extend these ideas to multi-stage models, modeling and solving these complex multi-echelon multi-period supply chains under data driven inventory policies require the use of our ROGS approach. In the next sections, we will present the details of modeling these data-driven inventory policies, the modified ROGS algorithm, and some computational studies.

4.2 Data-Driven Inventory Models

Single-Echelon N-Period Expiration

We first present a single-echelon model with random demand and perishable inventory. In this setting, inventory has a shelf-life of up to N periods, i.e. any item that arrived N periods ago and is still sitting on the shelf will be expired and therefore needs to be discarded. For each batch of inventory received, between periods 1 and N, a portion of leftover inventory will expire and be discarded while the rest will be carried over to the next period. Finally, in period N, all of the leftover inventory must be discarded. After reviewing the inventory at the beginning of each period, an ordering decision is made. If an order is placed, it arrives immediately. The products are arranged and replenished in a way that customers take the older items first (FIFO).

The decision maker aims to minimize total costs (4.1) which are the sum of inventory holding costs (h) for any unsold item at the end of a period and shortage penalty costs (v) for unmet demand. Unsold items at the end of period t consist of expired products (y_t^l) and inventory ($y_{t,i}$ with $i = 1, .., N$). Since we know when each item arrived, we keep track of inventory age and account for expiration. Unmet customer demands are difficult to track in retailing. Consequently, backlogging will not be allowed in our model.

We define the following parameters:

- $a_{j,t}$: exogenous "state" value j at time t
- d_t: demand in period t
- h: unit holding cost
- v: lost sales penalty cost
- w_i: waste factor (i.e., fraction of inventory that has to be discarded after age i)

And the following variables:

- x_j: coefficient for external "state" j when deciding order quantity
- q_t: order quantity at time period t
- $I_{t,i}$: age i inventory at the beginning of period t
- s_t: total units sold in period t

We formulate the problem as a MILP:

$$\text{DD-1R: } \min Z = \sum_{t=1}^{T} \left[\sum_{i=1}^{N} (hI_{t,i}) + v(d_t - s_t) \right] \tag{4.1}$$

$$\text{s.t. } S_t = \sum_{j=0}^{M} x_j a_{j,t} \qquad \text{for } t = 1, \ldots, T \tag{4.2}$$

$$q_t = S_t - I_t^b \qquad \text{for } t = 1, \ldots, T \tag{4.3}$$

$$s_t = \sum_{i=1}^{N-1} e_{t,i} \qquad \text{for } t = 1, \ldots, T \tag{4.4}$$

$$I_{t,1} = q_t \qquad \text{for } t = 1, \ldots, T \tag{4.5}$$

$$e_{t,i} \leq I_{t,i} \qquad \text{for } t = 1, \ldots, T; i = 1, \ldots, N \tag{4.6}$$

$$s_t \leq S_t \qquad \text{for } t = 1, \ldots, T \tag{4.7}$$

$$s_t \leq d_t \qquad \text{for } t = 1, \ldots, T \tag{4.8}$$

$$I_{t,i+1} = (1 - w_i)(I_{t-1,i} - e_{t-1,i}) \qquad \text{for } t = 1, \ldots, T; i = 1, \ldots, N \tag{4.9}$$

$$\hat{w}_{t,i+1} = l_i(I_{t-1,i} - e_{t-1,i}) \qquad \text{for } t = 1, \ldots, T; i = 1, \ldots, N \tag{4.10}$$

$$I_t^l = \sum_{i=1}^{N} \hat{w}_{t,i} \qquad \text{for } t = 1, \ldots, T. \tag{4.11}$$

$$I_t^b = \sum_{i=1}^{N} I_{t,i} \qquad \text{for } t = 1, \ldots, T. \tag{4.12}$$

The objective (4.1) is then to minimize the total cost over the planning horizon. The cost includes the holding cost of all inventory at all stages. Rather than explicitly model ordering cost, selling price, and salvage value, we use an equivalent formulation that assigns a shortage penalty cost for unsatisfied demand. We assume no loss of customer goodwill when shortages occur, though it can be easily incorporated into the penalty cost.

We assume that there is no inventory at the beginning of the first period, i.e. ($y_{0,i} = 0$). Because a stationary basestock policy is not optimal with perishable inventory, the ordering decisions are made under a state-dependent data-driven basestock policy (4.2). The order-up-to level S is then a linear function of M exogenous variables $a_{j,t}$ ($j = 0, .., M$) If this state-dependent basestock level is smaller than the inventory available at the beginning of the period, a new order q_t up to the base stock level is placed (4.3). The order quantity is therefore

To ensure that items are sold in the sequence of their arrival according to the FIFO policy, we distinguish between items sold from inventory that are leftover from previous periods and items that just arrived with the most recent order with constraints (4.4) through (4.6). The sales are denoted $(e_{t,i})$ with $i = 1$ being the sales from the most recent order. The sum of the sales of all ages add to the total sales s_t.

In each period, sales amount is limited to the inventory on hand (4.7) or the demand (4.8), whichever is lower. Demand is either satisfied from inventory, from the new incoming order or lost (no backordering). After demand is realized, inventory levels have to be updated at the end of each period. Inventory from the previous period and items that arrived with the new order are carried over to the starting inventory of the next period if they are neither sold nor expired. The new inventory levels are determined according to constraint (4.9) where w_i denotes the waste factor (i.e. the fraction of inventory that has to be discarded.) The amount of inventory that has to be discarded (y_t^l) is thus the sum of waste $(w_{t,i})$ from inventory of all ages (4.10), (4.11) Lastly, we inventory at the beginning of the next period is updated with constraint (4.12).

This model can be easily implemented and optimized by commercial solvers such as CPLEX with relatively small values of $N < 5$ and $T < 200$ and results in a better total cost than the same model under a fixed basestock policy. To model more closely to reality, we explore multi-echelon models with a single warehouse and multiple retailers in the floowing sections.

Single-Warehouse Two-Retailer

Problem Setting

Next, we consider a single-warehouse, two-retailer supply chain system where each retailer seeks to satisfy independent external demand. In each period, each retailer places an order to the warehouse and the warehouse will place an order to an external supplier assumed to always be able to satisfy any order quantity. The perishable inventory at each location has an expiration process similar to DD-1R. For simplicity, we assume for now that the age of the inventory does not carry over from the warehouse to the retailers (i.e. the inventory will expire in N days after each retailer receives it, regardless of how long it has been at the warehouse.). Further modeling of inventory expiration will be discussed in Section 4.6

Inventory Allocation

In the event that the warehouse is unable to satisfy the orders from the two retailers, it will require an implementable inventory allocation policy. It is important to note that, due to the nature of data-driven optimization, without an implementable allocation policy, the model optimizes in a deterministically optimal way for inventory to be allocated among the retailers. Such allocation decision is not implementable in the real world where the future is unknown to the policy decision maker.

One intuitive rule is to allocate inventory among the retailers proportional to the order amounts requested by each retailer. For example, if retailer 1 orders 10 units and retailer 2 orders 20 units and the warehouse has 15 units available, retailer 1 will receive 5 units and retailer 2 will receive 10 units. However, this would introduce a non-linear constraint to the model, as it is necessary to divide one decision variable over another. Alternatively, one could minimize the difference in shortages between the two retailers. In the same ordering example, then, the total shortage across the two retailers would be 15 units, hence, each retailer will be 7.5 units short from its original order. As a result, retailer 1 will receive 2.5 units and retailer 2 will receive 12.5 units. Using this approach, the model will remain a mixed integer *linear* program.

Model Formulation

To formulate this model, we index each stage of the supply chain where stages 1 and 2 represent the two retailers, and stage 3 represents the warehouse, as illustrated in Figure 4.1.

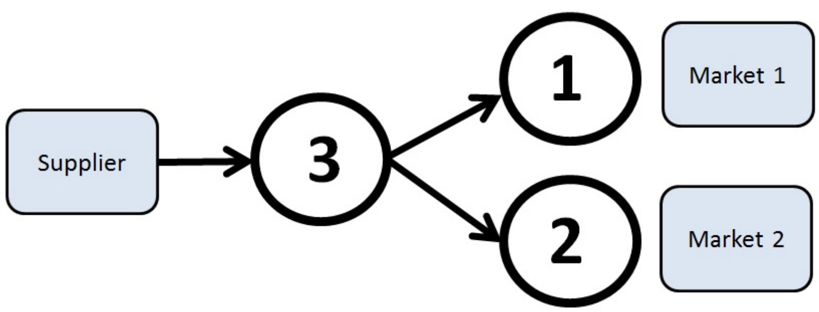


Figure 4.1: Single Warehouse Two Retailers

We introduce some new notation for the parameters and variables:

$$
\begin{aligned}
q_t^k &: \text{order Quantity from stage k at time t} \\
Q_t^k &: \text{ actual order amount received at stage k at time t, k = 1, 2} \\
c &: \text{ purchasing cost per unit ordered} \\
\widehat{I}_t^k &: \text{ inventory level at the beginning of period t at stage k} \\
\tilde{I}_t^k &: \text{ lost inventory at stage k at the end of period t } t \\
d_t^k &: \text{ demand in period t for market k, k = 1, 2} \\
a_{2,t}^k = p_t^k &: \text{ price in period t for market k,} \\
a_{3,t}^k = \hat{I}_{t-1}^k &: \text{ demand in period t-1 for market k,} \\
s_t^k &: \text{ sales in period t at stage k}
\end{aligned}
$$

The MILP is then formulated as the following:

$$
\text{DD-2R: min } z = \sum_{k=1}^{2}\sum_{t=1}^{T}\left[\sum_{i=1}^{N}\left(h^k I_{t,i}^k\right) + v(d_t^k - s_t^k)\right] + \sum_{t=1}^{T}\left[\sum_{i=1}^{N}\left(h^3 I_{t,i}^3\right) + \sum_{k=1}^{2} v(q_t^k - Q_t^k) + cQ_t^3\right] \tag{4.13}
$$

$$\text{s.t. } S_t^k = \sum_{j=0}^{M}(x_j^k a_{j,t}^k) \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.14}$$

$$q_t^k = S_t^k - \hat{I}_t^k \quad \text{for } t = 1, \ldots, T \tag{4.15}$$

$$s_t^k = \sum_{i=1}^{N-1} e_{t,i}^k \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.16}$$

$$I_{t,1}^k = Q_t^k \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.17}$$

$$e_{t,i}^k \leq I_{t,i}^k \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N; k = 1, \ldots, 3 \tag{4.18}$$

$$s_t^k \leq \hat{I}_t^k \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.19}$$

$$s_t^k \leq d_t^k \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.20}$$

$$I_{t,i+1}^k = (1 - w_i)(y_{t-1,i}^k - e_{t-1,i}^k) \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N; k = 1, \ldots, 3 \tag{4.21}$$

$$\hat{I}_t^k = \sum_{i=1}^{N} I_{t,i}^k \quad \text{for } t = 1, \ldots, T; k = 1, \ldots, 3 \tag{4.22}$$

$$I_{t,i+1}^k - e_{t,i}^k - z_{t,i}^k M \leq 0 \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N; k = 1, \ldots, 3 \tag{4.23}$$

$$z_{t,i+1}^k \leq z_{t,i}^k \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N; k = 1, \ldots, 3 \tag{4.24}$$

$$Q_t^1 + Q_t^2 \leq s_t^3 \quad \text{for } t = 1, \ldots, T \tag{4.25}$$

$$Q_t^k \leq q_t^k \quad \text{for } t = 1, \ldots, T; k = 1, 2 \tag{4.26}$$

$$Q_t^3 = q_t^3 \quad \text{for } t = 1, \ldots, T. \tag{4.27}$$

$$\tag{4.28}$$

The objective (4.13) then remains the same as model DD-1R for each stage. The penalty cost is charged at the warehouse level (stage 3) based on the order quantity requested by the retailers.

While we assume the external supplier will always satisfy any order placed by the warehouse, the warehouse may not always have sufficient inventory to satisfy the orders placed by the retailers. As a result, for stages 1 and 2 (i.e. retailers) the actual amount of inventory received, Q, may be less than the amount ordered, q. The specific rules for allocating limited inventory among retailers is discussed in Section 4.2. To ensure FIFO of the inventory (sell oldest first), we need to introduce constraints (4.23) and (4.24), where $z_{t,i}^k$ are binary variables and M is a large number. In each period, the warehouse cannot ship more than inventory on hand (4.38). We also need to make sure the retailers (stages 1 and 2) receive no more than the order amount (4.39). For the warehouse (stage 3), we assume all orders can be fulfilled by an external supplier (4.40).

For the allocation policy, we need to add the following set of constraints. Recall that we minimize the shortage differences between the retailers. This results in 3 cases, which require two binary variables, $\tilde{z}$ and ${z'}^k_t$:

- If retailer orders can be satisfied by warehouse inventory, namely $q_t^1 + q_t^2 \leq \hat{I}_t^3$, then $Q_t^1 = q_t^1$ and $Q_t^2 = q_t^2$

$$q_t^1 + q_t^2 \leq \hat{I}_t^3 + \tilde{z}_t M \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.29}$$

$$\hat{I}_t^3 \leq q_t^1 + q_t^2 + (1 - \tilde{z}_t) M \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.30}$$

$$Q_t^k + \tilde{z}_t M \geq q_t^k \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.31}$$

- If retailer orders cannot be satisfied by the warehouse inventory, but each retailer will receive some non-zero amount, namely $q_t^1 + q_t^2 \geq \hat{I}_t^3$ by an amount less than either $2q_t^1$ or $2q_t^2$, then each retailer will handle the same amount of shortage

$$q_t^1 + q_t^2 - \hat{I}_t^3 \leq 2q_t^k + {z'}^k_t M \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.32}$$

$$2q_t^k \leq q_t^1 + q_t^2 - \hat{I}_t^3 + (1 - {z'}^k_t) M \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.33}$$

$$Q_t^1 + ({z'}^1_t + {z'}^2_t + (1 - \tilde{z}_t)) M \geq q_t^1 - \frac{q_t^1 + q_t^2 - \hat{I}_t^3}{2} \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.34}$$

$$Q_t^2 + ({z'}^1_t + {z'}^2_t + (1 - \tilde{z}_t)) M \geq q_t^2 - \frac{q_t^1 + q_t^2 - \hat{I}_t^3}{2} \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.35}$$

- If retailer orders cannot be satisfied by the warehouse inventory, and that one of the retailers will receive nothing, namely $q_t^1 + q_t^2 \geq \hat{I}_t^3$ by an amount greater than one of $2q_t^1$ or $2q_t^2$, then the retailer which places a smaller order will receive zero, and the other will receive the entire $\hat{I}_t^3$

$$Q_t^k \leq 0 + (1 - {z'}^k_t) M \qquad \text{for } t = 1, \dots, T; k = 1, 2 \tag{4.36}$$

In a two-retailer setting, solving the MILP is much slower than the single echelon model, but remains practical with moderate values of $N < 3$ and $T < 100$. In the next section, we will build a model for a generic number of R retailers.

Single-Warehouse, Multiple-Retailer

Extending the two-retailer DD-2R model, we build a general model for any number of retailers. We represents a warehouse, stages $1, 2, ..., R$ represent retailers. Each stage may have a different holding cost while the inventory expiration days remain the same. We assume the age of inventory *does not* carry over from the warehouse to the retailers.

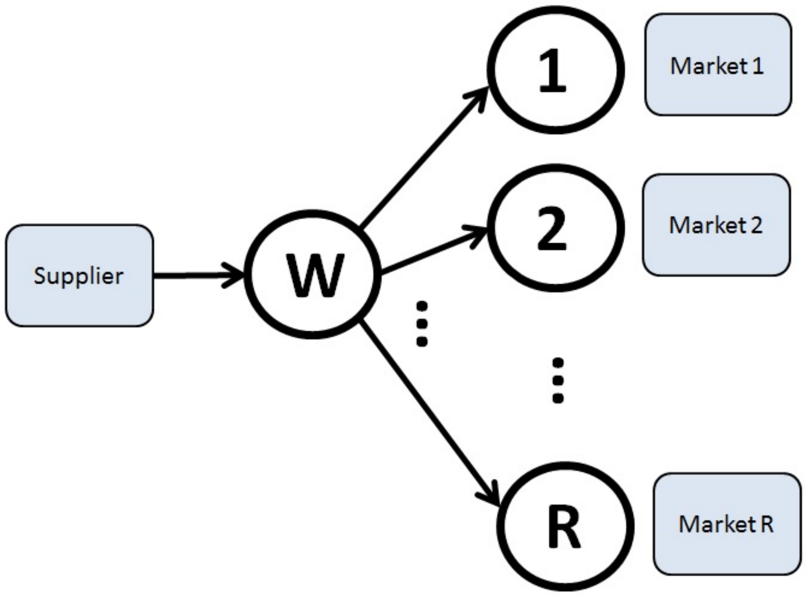


Figure 4.2: Single Warehouse R-Retailers

The MILP is then formulated as the following:

$$\text{DD-RR: min } z = \sum_{k=1}^{R}\sum_{t=1}^{T}\left[\sum_{i=1}^{N}\left(h^k I_{t,i}^k\right) + v(d_t^k - s_t^k)\right] + \sum_{t=1}^{T}\left[\sum_{i=1}^{N}\left(h^w I_{t,i}^w\right) + \sum_{k=1}^{R} v(q_t^k - Q_t^k) + cQ_t^w\right] \tag{4.37}$$

s.t. (4.14) - (4.24)

$$\sum_{k=1}^{R} Q_t^k \leq s_t^w \qquad \text{for } t = 1, \ldots, T \tag{4.38}$$

$$Q_t^k \leq q_t^k \qquad \text{for } t = 1, \ldots, T; k = 1, 2 \tag{4.39}$$

$$Q_t^w = q_t^w \qquad \text{for } t = 1, \ldots, T. \tag{4.40}$$

$$\sum_{k=1}^{R} q_t^k \leq \hat{S}_t^w + \tilde{z}_t M \qquad \text{for } t = 1, \ldots, T \tag{4.41}$$

$$\hat{S}_t^w \leq \sum_{k=1}^{R} q_t^k + (1 - \tilde{z}_t) M \qquad \text{for } t = 1, \ldots, T \tag{4.42}$$

$$Q_t^k + \tilde{z}_t M \geq q_t^k \qquad \text{for } k = 1, \ldots, R \; t = 1, \ldots, T \tag{4.43}$$

$$\frac{\sum_{\kappa=1}^{R} q_t^\kappa - \hat{S}_t^w}{R} \leq q_t^k + {z'}_t^k M \qquad \text{for } t = 1, \ldots, T; k = 1, \ldots, R \tag{4.44}$$

$$q_t^k \leq \frac{\sum_{\kappa=1}^{R} q_t^\kappa - \hat{S}_t^w}{R} + (1 - {z'}_t^k) M \qquad \text{for } t = 1, \ldots, T; k = 1, 2 \tag{4.45}$$

$$Q_t^k \leq q_t^k - \frac{\sum_{\kappa=1}^{R} q_t^\kappa - \hat{S}_t^w}{R} + {z'}_t^k + (1 - \tilde{z}_t)) M \qquad \text{for } t = 1, \ldots, T; k = 1, \ldots, R \tag{4.46}$$

$$Q_t^k \leq 0 + (1 - {z'}_t^k) + (1 - \tilde{z}_t)) M \qquad \text{for } t = 1, \ldots, T; k = 1, \ldots, R \tag{4.47}$$

$$\sum_{k=1}^{R} Q_t^k + (1 - \tilde{z}_t)) M \geq \hat{S}_t^w \qquad \text{for } t = 1, \ldots, T \tag{4.48}$$

The R-retailer DD-RR model is very similar to the two-retailer DD-2R model. The objective (4.37) now includes the cost of additional retailers. Constraints (4.14) through (4.24) are carried over from DD-2R while more constraints will be introduced for the inventory allocation policy. The first case, when the inventory at the warehouse is sufficient to satisfy all the retailers' orders, is enforced by constraints (4.41) through (4.43), where $\tilde{z}_t$ is binary. $\tilde{z}_t$ will be equal to 0 for each period where the warehouse has sufficient inventory to satisfy all orders from the retailers, and equal to 1 otherwise.

When the warehouse inventory is not sufficient to satisfy all orders (i.e. $\tilde{z}_t = 1$), we attempt to minimize the difference in shortages as before. This implies that the shortage will be divided evenly across all retailers. In other words, each retailer will experience a shortage of $\frac{\sum_{\kappa=1}^{R} q_t^{\kappa} - \hat{S}_t^w}{R}$. We will call this the "fair share" of shortage that each retailer should experience.

However, if one or more retailers place an order less than the "fair share" of shortage, they cannot take delivery of negative inventory. Therefore, the other retailers will need to experience additional shortage beyond their "fair share". This case does not scale well with increased number of retailers, namely, the number of integer constraints will grow exponentially as the number of retailers increase.

For each retailer, k, when $\tilde{z}_t = 1$ and ${z'}_t^k = 0$, retailer k will experience *at least* its "fair share" of shortage. Otherwise, that retailer will receive zero inventory. Specifically, each ${z'}_t^k = 0$ if retailer k orders more than the "fair share" of shortage (i.e. constraints (4.44) and (4.46) will be active and that retailer will receive non-zero amount of inventory), and equal to 1 otherwise (i.e. constraints (4.45) and (4.47) will be active and that retailer will receive no inventory). Lastly, we ensure that the warehouse must ship all its inventory to the retailers when there is a shortage (4.48).

This model presents a new challenge over the two-retailer DD-2R model. The additional constraints required to model an implementable inventory allocation policy, (4.44) through (4.48), introduce a large number of integer variables which require an enormous computational effort to solve. Analyses that rely on this MILP formulation alone will be impossible using traditional solvers like CPLEX. We will apply the ROGS algorithm to obtain decision variable values that work well for this model. However, we need to first modify our ROGS algorithm slightly to better suit this problem.

4.3 ROGS-DD: ROGS Algorithm for Data-Driven Models

To approximately solve the large MILP, DD-RR, the ROGS algorithm first solves a simplified version of DD-RR to obtain a starting point for the gradient search over the full DD-RR model. In the case of the multi-echelon data-driven models, what makes the MILP formulation difficult to solve is not only the length of the sample path, but also the constraints regarding the FIFO order of inventory consumption, (4.23) (4.24), and the constraints enforcing the allocation ratio among multiple retailers, (4.44) - (4.48). To address this, we modify the ROGS algorithm to first solve a relaxed version of the DD-RR that excludes constraints (4.23), (4.24), and (4.44) - (4.48). We then, through simulation, enforce these rules when evaluating the total cost during the gradient search step.

In the following outline of the general modified ROGS algorithm, we call the original problem P, in this case DD-RR, and the relaxed MILP $P^{'}$, in this case DD-RR without the FIFO and allocation constraints.

Algorithm (ROGS-DD):

- Step 0: Assign the parameter values of ROGS-DD
  1. Determine a sample path length, k, long enough for the objective function value of $P(\bar{X}_k)$ to capture the long run average cost of the supply chain system, and that $P^{'}(\bar{X}_k)$ can be relatively easily solved using a straightforward application of commercial optimization software.
  2. Determine initial gradient estimation and search step sizes, a and c.
  3. Determine the gradient search termination threshold δ
- Step 1: Solve $P^{'}(\bar{X}_k)$
  1. Formulate a relaxed MILP $P^{'}(\bar{X}_k)$.
  2. Solve the MILP to find the optimal vector of decision variables, θ'.
- Step 2: Solve $P(\bar{X}_k)$ using Gradient Estimation and Search
  1. Using the same sample path, $\bar{X}_k$, and formulate an MILP $P(\bar{X}_k)$.
  2. Build a simulation to evaluate $P(\bar{X}_k)$ with any given decision variables, θ
  3. Use θ' as a starting point for finite difference gradient estimation and search. At each iteration, the gradient estimation and step sizes are reduced following the sequences $a_n = an^{-1}$ and $c_n = cn^{-1/6}$.
  4. Repeat the search until the estimated gradient in all directions is less than δ. Return the decision variable vector, θ^*

Simulation Algorithm for Inventory Allocation

In our simulation, we determine how inventory is allocated between the retailers using a sorting and elimination algorithm in place of the integer variable and constraints (4.44) through (4.48). For each period where the warehouse cannot satisfy all retailers' orders, we minimize the difference in shortages between the retailers. Recall that a retailer cannot experience a shortage greater than that of its original order quantity. We first sort the order sizes from small to large, then we sequentially eliminate the retailers with orders that are too small to receive any inventory at all, starting with the smallest order and working our way up. We repeat this until all remaining retailers will receive a non-zero amount of inventory. These remaining retailers will then experience the same shortage in that period. Specifically, in each period, the warehouse will divide its inventory, $\hat{S}^w_t$ among all the retailers with the following algorithm:

- If $\hat{S}^w_t > \sum_{\kappa=1}^{R} q^\kappa_t$:
  Let $Q^k_t = q^k_t$ for all k.
- Else:
  Sort all orders q^k_t over all k. WLOG, let $q^1_t < q^2_t < ... < q^R_t$
  Let $shortage = \sum_{\kappa=1}^{R} q^\kappa_t - \hat{S}^w_t$.
  Let $r = 1$
  Begin loop:
  1. If $q^r_t > \frac{shortage}{R-r+1}$:
     For all $k < r$: $Q^k_t = 0$
     For all $k \geq r$: $Q^k_t = q^k_t - shortage$
     End loop
  2. Else:
     $shortage \leftarrow shortage - q^r_t$
     $r \leftarrow r + 1$
     Repeat loop

With this simulation, we can first evaluate the cost of the supply chain operation, using the heuristic solution obtained by solving P', under the warehouse distribution policy to minimize the difference in shortages between retailers. We will then follow the modified ROGS algorithm, using the heuristic solution as the starting point, to perform local gradient search to improve the state dependent inventory policy parameters.

4.4 Experimental Design

We will use the same general methodology as Section 3.2 to test our state-dependent data-driven models and the ROGS-DD algorithm. For our experiments, we will focus on the single-warehosue multi-retailer model. We begin our experiments with a 3-retailer model with sythetic data where we generate our 100-period ($k = 100$) retrospective optimization sample paths with pairs of correlated demand and price in each period. We let the holding costs (h) be the same across all stages, and set the backorder penalty cost of 9 per unit ($v = 9$). We then take this sample path and formulate the integer program as described in the previous section to solve for the optimal policy parameters. We experiment with the different combinations of the following values of $h = 2, 4, 6, 8, 10$

We test each combination of these values on 100 randomly generated sample paths and collect the data from each run. We then evaluate the performance of these supply chain settings over 100 randomly generated sample paths of 900-periods each. We report the average cost from these 100 sample paths. In addition, we compare the total cost and service level over the 900 periods using our parameter settings versus ones obtained by the traditional basestock setting.

Recall that the inventory allocation policy does not scale well with increased number of retailers in MILP modeling. To overcome this issue, we will first relax this constraint, and instead enforce a bound on each retailers' maximum shortage. We will then use this relaxed MILP in the first step of our ROGS-DD algorithm. In the second step of the ROGS-DD algorithm, we use the simulation model outlined in Section 4.3 to fully capture our intended inventory allocation policy.

4.5 Experiment Results and Discussion

3-Retailers with Generated Data

Effectiveness of ROGS-DD

To demonstrate the effectiveness of ROGS-DD with state-dependent inventory policy, we compare the solutions under a stationary basestock policy in contrast to our state-dependent policy. The stationary basestock policy is found by replacing constraint (4.14) by a single variable representing the order-up-to level. In addition, we compare our model against solutions obtained by different relaxed versions of the state-dependent inventory model. In one case, we relax all the distribution policy constraints; in the other, we further relax the FIFO constraint in our model. We first obtain decision variables from these 4 solution approaches using 100 periods of randomly generated demand and other data. We then evaluate the total cost over the next 900 periods and compare the results across these approaches.

Due to the solution times of these more complex MILP's, generally requiring 6 to 12 hours, we replicate this experiment over only 10 independently generated sample paths. We then evaluate each of the above solutions over a randomly generated 900-period sample path with all FIFO and distribution constraints. We present the average total cost obtained by these approaches in Table 4.1 below:

Table 4.1: Holding Cost = 2

Solution Approach	Total Cost	Service Level
Basestock Policy	360527	98.75%
Relaxed MILP w/o FIFO	295508	98.95%
Relaxed MILP w/ FIFO	262156	97.63%
ROGS-DD	243522	97.43%

From Table 4.1, we can see that ROGS-DD is able to obtain a much lower total cost over a stationary basestock policy. Furthermore, we observe that the service level reduces only slightly. While the service level isn't part of the modeling objective or constraints, this shows that the data driven approach is able reduce costs by ordering less inventory on the days when less demand occurs and vice versa on days with higher demand. On the other hand, a stationary basestock policy tends to hold too much or not enough inventory as demand fluctuates from day to day. While the solution times for the non-relaxed MILP's often take 6 to 12 hours to solve, ROGS-DD can obtain a solution in 1 to 2 hours.

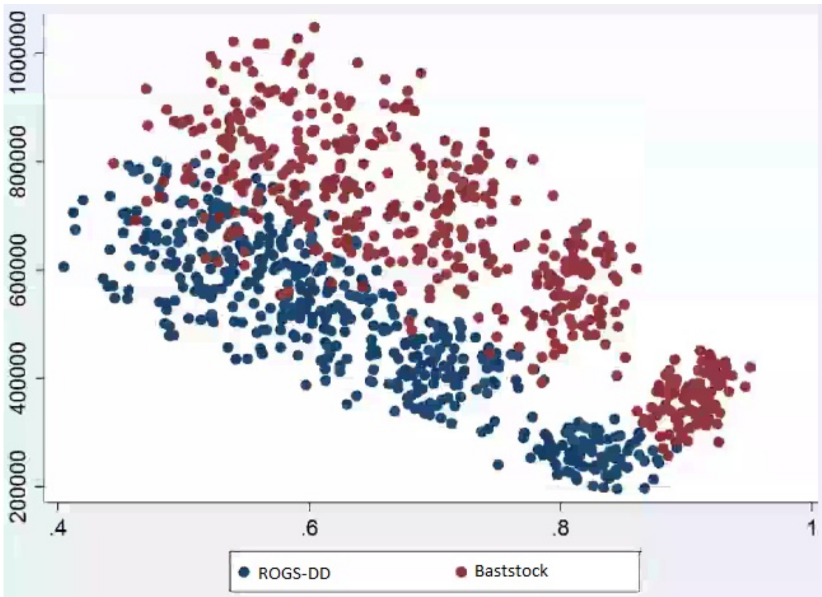
Holding Costs and Service Levels

We further test our model over a set of different holding costs.

Table 4.2: Single-Warehouse 3-Retailer

Holding Cost	Basestock Cost	ROGS-DD Cost	Basestock Service	ROGS-DD Service
$h = 2$	362492	253092	98.36%	96.95%
$h = 4$	563169	408213	96.18%	94.07%
$h = 6$	689073	514436	93.37%	91.53%
$h = 8$	773414	596022	90.23%	89.45%
$h = 10$	842994	663950	87.30%	87.32%

Figure 4.3: Single-Warehouse 3-Retailer



From Table 4.2 we observe that in all cases of various holding costs, ROGS-DD is able to provide significant improvement on the total cost with little trade-off in service level. Figure 4.3 plots the service level against total cost over the 500 runs and we can see clearly that the state-dependent inventory policy solved by ROGS-DD outperforms the standard stationary basestock approach.

3-Retailers with Real World Data

To test our model on real world data, we obtained data from a leading German grocery chain store over 2000 days. In general, a state-dependent inventory policy is more effective when the random variable, such as demand, in the system is correlated with some observable exogenous factors. Therefore, we analyze the correlation between the daily sales and various factors, including selling price, temperature, day of the week, etc. as shown in Figures 4.4, 4.5 and 4.6. We find that the sales amount does not correlate with each particular factor very strongly, while it may correlate with a combination of multiple factors. Further, we find that the daily sales amount are highly correlated with the sales incurred in the previous day. We conjecture that the demand is not independent from day to day. Therefore, we decide to incorporate the previous day demand as one of the exogenous state variables in our model.

Figure 4.4: Sales vs. Selling Price

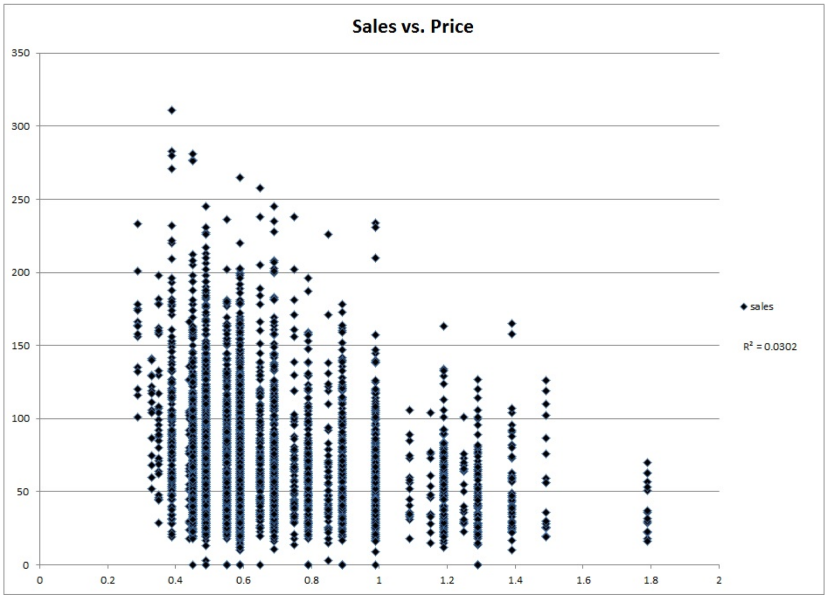


Figure 4.5: Sales vs. Daily Temperature

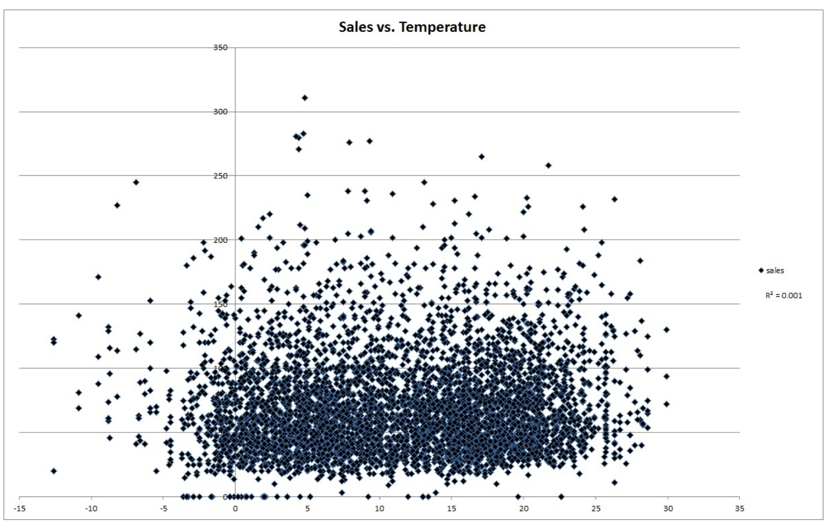
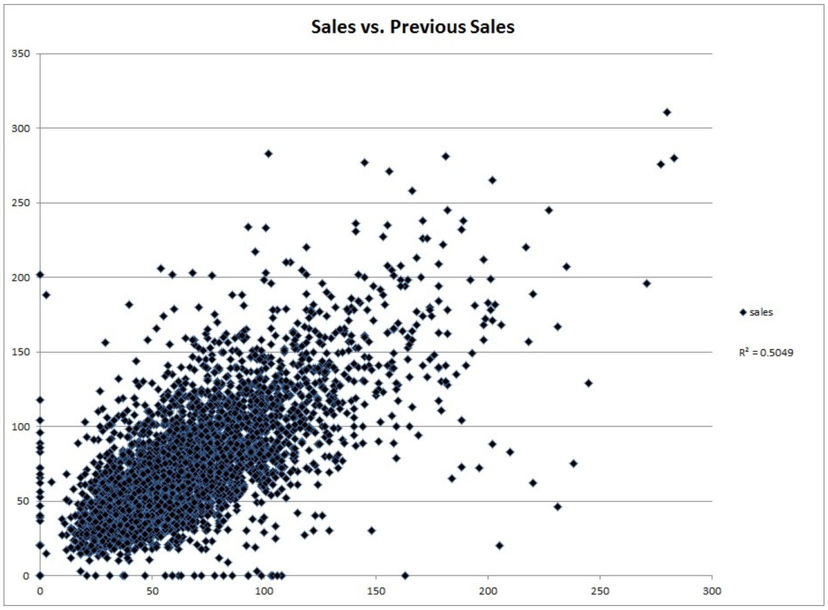


Figure 4.6: Sales vs. Previous Period Sales



In addition, the stores currently do not operate on Sundays. We examine different sets of assumptions when applying real world data to our model:

- "Previous demand" data for Monday can refer to either Saturday's demand or Sunday's demand (zero).
- Inventory can either age by 1 day from Saturday to Monday, or by 2 days from Saturday to Monday.

In reality, "previous day's demand" for Mondays will always be zero. However, we see that each Saturday's demand is moderately correlated with the demand on the Monday that follows. We conjecture that using Saturday's demand data to decide the order quantity on Monday will result in a more effective inventory policy than using zero value. If so, we can modify our data input as if that there are only six days in a week and modify our model to reflect an additional day of aging once every six days by changing the waste factors. We test these different assumptions and observe the results.

Furthermore, we modify our state-dependent data driven model to incorporate all the data that are correlated with the demand. This includes a total of 8 exogenous state ($M = 8$) variables, namely:

- Demand in the previous period

- Current period selling price
- Current period temperature
- Binary variable for each day of the week
  e.g. For Monday: $Mon = 1, Tue = 0, Wed = 0, Thur = 0, Fri = 0$

We also experiment with two sets of shelf lives:

- 3-period expiration with waste factors $w = [0.0, 0.1, 1.0]$
- 5-period expiration with waste factors $w = [0.0, 0.2, 0.8, 0.95, 1.0]$

We take the first 200 periods of data to formulate our data driven MILP, P. As before, we first relax the distribution rule and FIFO rule to form P' and use CPLEX to obtain a solution. Then, we use ROGS-DD to improve the solution on P. Finally, we evaluate the total cost using the next 1400 periods of data.

We first test each set of assumptions and evaluate the total cost of the data-driven inventory policy. Table 4.3 shows the results of different sets of assumptions:

Table 4.3: Holding Cost = 2

Assumptions	Aging skip Sunday	Aging Include Sunday
"Previous" for Monday = Saturday	542768	500128
"Previous" for Monday = Sunday (0)	639314	639950

With the above results, we conclude that our conjecture about demand correlation is correct. From this point forward, we will model using Saturdays' demand data as the "demand in the previous period" for Mondays. Table 4.4 lists the results:

Table 4.4: 3-Retailer Real World Data

Shelf Life	Total Cost P'	Total Cost ROGS-DD	% Improvement
3 period	477346	401954	15.8%
5 period	518469	428064	17.5%

Table 4.5: Coefficients for 3-Period Expiration

External Data	Constant	Price	Previous Sales	Temperature	Mon	Tue	Wed	Thur	Fri
Retailer 1	91.58	-51.83	0.5600	-0.1880	3.350	-32.57	-20.67	-6.787	5.536
Retailer 2	198.78	-109.98	0.4100	-2.698	17.49	-33.82	-32.46	7.204	19.93
Retailer 3	129.50	-52.82	0.5172	-0.5903	4.890	-57.60	-46.21	0.7462	7.680
Warehouse	184.17	-89.85	0.9075	-1.132	14.32	-68.84	-38.63	5.229	7.016

Table 4.6: Coefficients for 5-Period Expiration

External Data	Constant	Price	Previous Sales	Temperature	Mon	Tue	Wed	Thur	Fri
Retailer 1	93.30	-43.13	0.5566	-0.5390	-5.864	-40.02	-32.08	-16.54	2.585
Retailer 2	173.50	-76.64	0.3885	-1.9998	4.010	-44.70	-36.60	-4.279	16.055
Retailer 3	126.01	-43.62	0.4807	-0.3760	-2.431	-61.07	-50.38	-16.828	-2.900
Warehouse	144.20	-63.85	0.8018	-1.2350	1.526	-67.14	-31.16	2.037	5.722

From Table 4.4, we see that ROGS-DD is able to improve the total cost over 800 periods over the data-driven inventory settings obtained from the relaxed MILP. Tables 4.5 and 4.6 shows the coefficients corresponding to those costs. We can observe that the demand is positively correlated with some days of the week and negatively correlated with other days. Additionally, the demand also seems strongly negatively correlated with the selling price as expected.

65 Retailers with Real World Data

We further apply our model and solution algorithm to the large set of data we obtained from a real world grocery chain. We use data for one single product over 65 retailers and one single warehouse as our model parameters. Since each retailer sets its own ordering policy parameters, we now have nearly 600 decision variables in our model. This greatly increases the required computation time of finite difference gradient estimation and search. Unfortunately, simply increasing the gradient search step sizes does not show significant improvement in either the computation time or the solutions obtained. We then decide to simultaneously perturb multiple directions in each gradient search iteration in order to reduce the number of gradient estimations required to reduce the total cost. Details of this modification in our algorithm will be discussed later in this section. It is also important to examine the cost-effectiveness of additional runtime beyond a certain duration as the gradient search stepsize can reduce to an insignificant amount after a large number of iterations.

Computational Time with Finite Difference Gradient Estimation and Search

Our ROGS-DD algorithm uses a finite difference approach in the gradient search step. Because we now have nearly 600 decision variables, to estimate the finite difference in each possible pair of combinations of those variables require a huge amount of computation time. To thoroughly examine this challenge, we let the ROGS-DD algorithm run with different time limits and see how much improvement it can achieve. Table 4.7 shows the results:

Table 4.7: ROGS-DD 1-direction

Time limit	Total Cost P'	Total Cost ROGS-DD	% Improvement
24 hrs	1,699,958	1,699,378	0.034%
72 hrs	1,699,958	1,561,361	8.15%

As seen in Table 4.7, only very little improvement can be achieved by running the gradient search for 24 hours. While a 72 hour runtime shows more significant improvement. This result suggests that the algorithm, while slow, remains to be effective in improving total operational cost. The challenge is to improve the computational efficiency of ROGS-DD.

Effectiveness of Simultaneous Perturbation

To improve computational efficiency of ROGS-DD, we perturb more than one decision variable in each iteration of the gradient search, thereby achieving further improvement in the objective function value. In each iteration, we first rank the estimated gradient in each direction obtained by finite difference estimation. We then simultaneously perturb the n steepest gradient using the next n reduced step sizes. With this approach, any perturbation direction that causes the objective function to *increase* as a result of being simultaneously perturbed with other variables can be corrected in the following iteration, while reducing up to n times the computational effort. Table 4.8 compares the performance of simultaneously perturbing $n = 1, 3,$ and 5 directions in each gradient search iteration. In each of these experiments, the computation time is limited to 24 hours.

Table 4.8: ROGS-DD Number of Perturbations

# Directions	Total Cost $P^{'}$	Total Cost ROGS-DD	% Improvement
1 Direction	1,699,958	1,699,378	0.034%
3 Directions	1,699,958	1,532,623	9.84%
5 Directions	1,699,958	1,527,381	10.15%

Runtime of Simultaneous Perturbation

We then examine how much time we should allow the ROGS-DD algorithm to run. We fix the number of simultaneous perturbations to 5 directions and let the program run for various times. Table 4.9 shows the results:

Table 4.9: ROGS-DD 5-directions

Time limit	Total Cost $P^{'}$	Total Cost ROGS-DD	% Improvement
12 hrs	1,699,958	1,532,391	9.86%
24 hrs	1,699,958	1,489,197	12.40%
48 hrs	1,699,958	1,479,077	12.99%

From Table 4.9, we observe that the gradient search algorithm is able to provide significant cost improvement in the first 24 hours. The effectiveness of this, however, diminishes as the gradient search step sizes reduce over time. In the next section, we use this insight when we apply ROGS-DD to solving the state-dependent data-driven model with data from 65 retailers.

65-Retailer Model Results

From Table 4.9, we observe that using the 5-direction simultaneous perturbation methodon a 65-retailer model, the effectiveness of gradient search reduces drastically when the step sizes reduce beyond 24 hours of runtime. With a daily holding cost of 2 and penalty cost of 9, our model optimizes over a 200-period in-sample data set and evaluates the total cost with a 1400-period out-of-sample data set over 65 retailers and one warehouse. Table 4.10 shows the solutions obtained by the relaxed data-driven MILP, $P^{'}$, and ones obtained by the ROGS-DD algorithm. Table 4.11 shows the coefficients obtained by ROGS-DD:

Table 4.10: 65-Retailer Real World Data Total Cost

Data	Total Cost P'	Total Cost ROGS-DD	% Improvement
200-period In-Sample	1,699,958	1,489,197	12.4%
1400-period Out-of-Sample	11,984,305	10,260,180	14.4%

Table 4.11: 65-Retailer Real World Data Coefficients

Retailer	Constant	Price	Prev Dem	Temperature	Mon	Tue	Wed	Thur	Fri
1	133.3530421	-93.286	0.306798	-0.591181	-16.1211	-12.8271	-15.478	15.1627	11.6104
2	152.897826	-61.6464	0.423691	-1.70419	-40.2015	-24.925	-26.3266	-4.38932	37.9687
3	100.4106667	-37.368	0.432142	0.953682	-30.4163	-32.0687	-32.1419	13.6256	17.8849
4	96.97063518	-42.7388	0.389754	0.312481	-29.075	-21.1901	-17.8481	21.8597	29.9269
5	120.6900029	-40.1321	0.42853	-0.00903679	-47.3554	-45.8365	-35.849	-2.57152	10.5886
6	76.52925934	-30.1918	0.63903	-0.259447	-0.770446	-4.59955	-5.90525	32.1156	52.3875
7	123.4303333	-62.2179	0.525919	0.283544	-9.14965	0.65962	-21.6603	19.9122	42.8756
8	136.9661899	-54.4898	0.291837	-1.84694	-39.8296	-17.7765	-25.8704	-14.1561	23.2694
9	72.73243586	-27.1139	0.453611	0.235047	-24.6521	-17.9009	-16.8305	20.11	27.7708
10	124.4695441	-38.8755	0.183219	1.12463	-62.3283	-47.9307	-58.9757	-52.5332	2.01088
11	102.625	-53.8277	0.433589	0.0225064	-15.7599	-21.4083	-13.8842	7.4544	13.9404
12	107.3809042	-25.5556	0.482105	0.201835	-32.5093	-38.5688	-34.7358	16.7867	46.3872
13	128.616378	-60.5687	0.394501	-1.43678	-0.606346	-15.1764	-27.0776	14.7058	15.9824
14	138.7352295	-68.1266	0.273061	-0.992839	9.99774	-22.925	-22.9831	9.43001	20.3655
15	107.9555039	-44.1622	0.469251	-0.417769	-31.1864	-35.4772	-31.5719	0.342908	4.9576
16	168.806	-64.9884	0.268353	-0.634237	-57.7112	-41.0863	-59.4851	-36.189	44.325
17	82.60705981	-35.3373	0.387368	-0.285382	-23.7011	-23.6946	-21.8748	-0.12444	6.85937
18	98.72606347	-51.0786	0.602394	-0.080944	-26.0378	-27.5796	-38.1731	6.56199	15.7975
19	56.8554	-32.73009192	0.432377	-0.10298	-1.97323	-3.14802	-0.113415	18.5967	16.448
20	145.2587215	-11.8729	0.731069	2.40729	-153.923	-95.0907	-109.262	-53.7839	58.1089
21	147.2432241	-43.6949	0.527456	-0.620155	-74.5771	-58.8865	-61.4454	-12.2981	18.7425
22	177.2353333	-52.5029	0.601488	-0.653593	-104.933	-87.9079	-87.7261	-37.188	5.79295
23	198.3983951	-97.2052	0.335299	-0.908943	-24.5236	-25.6193	-39.3065	-11.4837	5.30901
24	118.525889	-47.3552	0.438511	0.270809	-40.0224	-49.8192	-37.346	-12.7224	-14.6623
25	133.9929291	-59.0728	0.490353	-1.27526	-33.1669	-34.2402	-30.4829	-12.4686	-0.764811
26	205.8463279	-94.7361	0.18054	-1.83192	-25.965	-41.701	-41.4433	-24.2826	10.4576
27	104.7145955	-47.332	0.596921	0.43778	-34.856	-28.7281	-23.2076	14.7117	19.1072
28	133.9326552	-50.4317	0.296104	0.450828	-26.5737	-27.2377	-34.6742	-5.57024	2.45849
29	66.94736154	-25.8867	0.558739	1.21271	-27.2752	-12.5409	-21.4094	-2.73963	15.5
30	84.85107001	-36.1519	0.414061	-0.367309	-8.20264	-22.2176	-20.1677	23.4562	34.0069
31	133.2912781	-56.4094	0.275873	-0.435537	-39.5665	-12.2055	-40.4071	0.420384	58.6515
32	124.4112257	-57.4798	0.324328	-0.127264	-25.8327	-26.0691	-30.3762	5.36924	10.4464
33	109.123069	-41.0978	0.378284	-0.502872	-43.4463	-17.5793	-25.8577	-3.09795	15.0308
34	147.0286578	-48.9847	0.509651	-0.189736	-77.253	-51.6674	-59.1019	-10.3458	37.6622
35	94.19631057	-22.3636	0.160287	0.187795	-47.1001	-28.9082	-33.6989	-28.4605	15.8616
36	121.202102	-22.4951	0.238244	-0.790265	-77.6822	-62.139	-59.7871	-50.9907	14.4462
37	94.70625726	-41.202	0.209193	-0.216235	-15.5793	-12.4279	-22.0896	10.9829	14.605
38	81.63370841	-39.8335	0.594546	-0.327443	-26.7388	-11.6341	-11.8328	15.4546	13.3577
39	179.1946432	-76.2223	0.145124	-0.996441	-23.1145	-35.5507	-45.2102	-5.82699	4.72065
40	133.966	-54.2825	0.288136	0.0338983	-29.0912	-38.4708	-25.7172	-19.3285	-6.16009
41	70.83961101	-22.4573	0.421414	-0.929589	-12.7181	-13.1247	-13.0076	7.8807	17.7325
42	77.57902098	-32.3394	0.30948	-0.120795	-23.2937	-20.9927	-18.0988	-11.1257	5.8104
43	181.6439441	-71.4997	0.411715	-1.16983	-74.591	-59.7961	-52.3315	-0.31099	11.5452
44	118.8913391	-48.4666	0.524312	-1.40459	-43.0074	-23.7043	-26.7636	9.81613	21.201
45	49.4305295	-22.181	0.5961	-0.00111534	-9.30163	-9.49327	-7.4808	39.3828	45.8266
46	50.55742481	-2.61958	0.412029	1.18157	-28.3345	-22.739	-20.2846	4.02977	8.2277
47	128.9431914	-73.034	0.450986	0.542756	-5.4323	2.23217	-14.9693	15.7457	35.5082
48	89.14200126	-45.7914	0.393243	-0.5825	-6.33932	-11.5673	-14.5961	13.7638	20.5182
49	141.4570378	-58.3662	0.406714	-1.03592	-49.2045	-28.2871	-29.5007	3.19759	5.49925
50	132.6560977	-56.6414	0.371228	-0.00285873	-33.4119	-35.3414	-37.1075	-9.5441	5.56652
51	250.1211543	-78.7681	0.176509	-1.52978	-103.161	-96.1227	-91.4332	-68.8185	90.5349
52	176.1619078	-69.7857	0.189354	0.373512	-40.0636	-42.7724	-41.3574	1.32408	1.73516
53	188.6447036	-89.2853	0.34041	-2.58164	-53.4251	-47.7498	-29.6487	-3.60125	22.7161
54	146.62282	-60.6282	0.509204	-0.643072	-68.0202	-52.6114	-58.0247	-16.6152	-23.9743
55	76.94164179	-15.9483	0.699355	0.3632	-45.4568	-20.7218	-8.50199	31.1881	31.2212
56	67.71377348	-38.6772	0.472043	0.139107	4.14761	-0.338436	-4.95843	28.9946	23.0863
57	92.82946005	-48.1208	0.537612	-0.561532	-19.5142	-9.83081	-16.9119	6.26795	20.5667
58	72.73086285	-29.6742	0.714828	-0.774104	-4.64674	18.5505	-5.77979	41.5735	35.3558
59	85.09167089	-24.058	0.518322	0.349407	-46.3444	-9.28022	-23.02093	-22.2572	12.0098
60	140.3541423	-37.6143	0.404188	0.615051	-90.1193	-76.5175	-72.6409	-39.4316	47.2249
61	82.76851557	-32.157	0.511959	-0.670357	-35.0463	-18.492	-21.7114	-12.9357	21.6433
62	178.8989295	-61.0172	0.229682	-1.53278	-71.6634	-58.2254	-65.1683	-53.8925	21.792
63	149.5820986	-59.4347	0.445533	-0.379091	-51.217	-40.6948	-57.3066	-24.0429	0.364055
64	92.10739622	-20.4115	0.190332	-0.812115	-31.7099	-23.9993	-28.5604	-33.952	-1.8578
65	144.8970392	-47.389	0.06127	-0.197202	-56.7877	-53.7735	-54.41	-41.5703	1.48781
Warehouse	36.3432	-16.7847	0.700248	0	-21.33333333	-8.83385	-18.54164103	44.26429851	16.8164

From Table 4.10, we see that the state-dependent data-driven inventory model DD-RR works well for setting effective inventory policy parameters for this grocery chain store. While the MILP is highly complex and almost impossible to solve using CPLEX or other existing solvers, our modified ROGS-DD solution approach is able to provide a good solution within a reasonable computation time.

4.6 Extensions

So far we have assumed that the aging process of the inventory does not carry over from the warehouse to the retailers. Namely, the retailer discards some fraction of the goods regardless of how long those goods have been stored in the warehouse level prior to being shipped to the retailers. We have begun modeling extensions to our models to incorporate different assumptions on the aging process. We are confident that our ROGS approach can be easily modified to adapt to these different, more complex models.

N-Period Two-Retailer Model with Cumulative Age

Extending DD-RR, this model adds the carryover of the age characteristic from the warehouse to the retailers. Each stage may have a different holding cost while the inventory expiration days remain the same. The age of inventory will carry over from the warehouse to the retailers. Figure 4.7 illustrates a 3-period expiration model ($N = 3$) with 2 retailers.

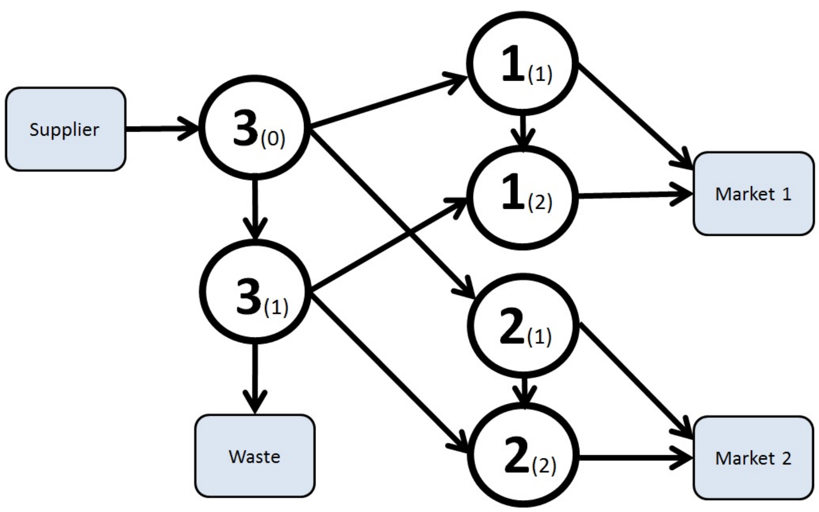


Figure 4.7: N-Period Two-Retailer Model with Cumulative Age

We modify some notation:

$\hat{q}_t^k$: total Order Quantity from stage k at time t

$q_{t,i}^k$: order Quantity from stage k at time t satisfied by age i inventory

$Q_{t,i}^k$: actual order amount received at stage k at time t, k = 1, 2 with age i

Our objective function remains the same as the previous models DD-1R, DD-2R, and DD-RR,and the constraints for the most part will be carried over as well, with the exception of the following:

$$I_{t,i+1}^k = (1 - l_i)(I_{t-1,i}^k + Q_{t-1,i}^k - e_{t-1,i}^k) \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N; k = 1, \ldots, 3 \tag{4.49}$$

For stages 1 and 2, we need to make sure we receive no more than the order amount. We also ensure we receive no more than the amount stage 3 can fulfill. For stage 3, we assume all orders can be fulfilled by an external supplier with zero-aged inventory.

$$Q_{t,i}^1 + Q_{t,i}^2 \leq e_{t,i}^3 \quad \text{for } t = 1, \ldots, T; i = 1, \ldots, N \tag{4.50}$$

$$Q_{t,i}^k \leq q_{t,i}^k \quad \text{for } t = 1, \ldots, T; k = 1, 2; i = 1, \ldots, N \tag{4.51}$$

$$Q_{t,0}^3 = q_{t,0}^3 \quad \text{for } t = 1, \ldots, T. \tag{4.52}$$

$$Q_{t,1}^3 = 0 \quad \text{for } t = 1, \ldots, T. \tag{4.53}$$

N-Period Two-Retailer Model with Different Expiration Processes

This model allows a different aging process at different stages. Each stage may have a different holding cost and the inventory expiration days may be different at each stage. The age of inventory will carry over from the warehouse to the retailers according to some equivalence function.

Figure 4.8 illustrates a 3-period aging model (N=3) where the inventory ages twice as fast at the retailers as it does at the warehouse.

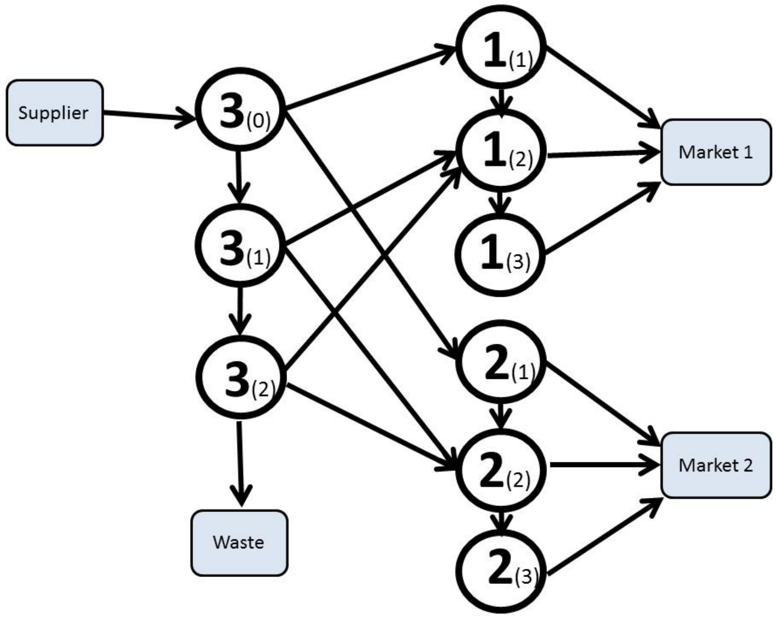


Figure 4.8: N-Period Two-Retailer Model with Different Expiration Process

With this model, we will then need to introduce an aging equivalence function:

$f(i)$: inventory with age i at retailer is equivalent to inventory with age f(i) at supplier
$f^{-1}(i)$: The inverse function of f

Then the constraint for received inventory at the retailer level would become:

$$I^k_{t,i+1} = (1 - l_i)(I^k_{t-1,i} + Q^k_{t-1,f^{-1}(i)} - e^k_{t-1,i}) \qquad \text{for all } t, i, N \tag{4.54}$$

While this formulation has not yet been tested, it should, in theory, allow for flexible modeling of a variety of different inventory aging and expiration processes.

4.7 Conclusion

Stochastic supply chain systems can be optimized using a data-driven approach where the models are built around historical data, and future operational policy decisions are made by optimizing these models. These supply chain models turn out to be highly complex mixed integer linear programs similar to the ones found in Retrospective Optimization. This allows us to apply our ROGS algorithm with little modification to significantly reduce computation times required to solve these models. Furthermore, ROGS allows for complex modeling via simulation in place of MILP modeling. The numerical studies have been successful and we are confident that ROGS-DD can be applied on other more complicated data-driven optimization models.

Chapter 5

Summary

In preliminary experiments, we found Retrospective Optimization to be an effective approach for optimizing supply chain systems. However, for more complex supply chain systems that include a multi-echelon structure, distribution or assembly phases, and random factors including demand, lead time, and disruptions, the Retrospective Optimization approach requires solving relatively large mixed integer linear programs that require extremely long computation times. To overcome this challenge, we developed the Retrospective Optimization and Gradient Search (ROGS) algorithm (3.1) to provide a good approximate solution to these retrospective MILP's. Our algorithm is efficient, effective, and is flexible enough to be applied to a wide variety of settings without significant modification, as shown in Section 3.2. In particular, we examine the application of ROGS to stochastic supply chain optimization problems under a Retrospective Optimization approach and a data-driven approach. The prior approach seeks effective policy parameter settings by building a model around simulated sample paths of the random variables in the system, while the latter does so by building a model around historic data of the random variables. While the approaches differ in principal, the resulting MILP model from these approaches present very similar challenges in complexity and solution time.

Using our ROGS algorithm, we are able to determine effective inventory policy parameters under a variety of supply chain settings, including multi-echelon systems, distribution networks, random lead times, demand, and disruption. The results obtained by ROGS is also shown to be more effective in minimizing overall operational cost than any other existing approaches. The ROGS algorithm also allows the use of simulation modeling for highly complex supply chain systems where mixed integer programming models become impractical. To demonstrate this, we apply ROGS and obtain effective policy parameters for a multi-echelon supply chain system with perishable inventory in Section 4.3.

While the ROGS algorithm is shown to be effective in solving supply chain problems, this general approach can be applied to many optimization problems with a mixed integer programming models. ROGS shows that while gradient search approaches in general do not work on non-convex functions, these techniques can remain effective when provided with a good set of starting point solutions and effective gradient search step sizes. In our experiments so far, the ROGS algorithm has been implemented using Python programming language in combination with the CPLEX solver. In the future, ROGS has the potential to be embedded into commercial supply chain optimization software with the development of a friendly user interface. While ROGS is not without its limitations, we are able to push its boundaries by adjusting some algorithm parameters and we intend to further explore these limitations in future research.

Bibliography

[1] Sridhar Bashyam and Michael C Fu. Optimization of (s, s) inventory systems with random lead times and a service level constraint. *Management Science*, 44(12-part-2):S243–S256, 1998.

[2] Dimitris Bertsimas and Aurélie Thiele. A Robust Optimization Approach to Inventory Theory. *Operations Research*, 54(1):150–168, 2006.

[3] Anna-Lena Beutel and Stefan Minner. Safety stock planning under causal demand forecasting. *International Journal of Production Economics*, 2011.

[4] M. Broadie, D.M. Cicek, and A. Zeevi. An adaptive multidimensional version of the kiefer-wolfowitz stochastic approximation algorithm. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 601–612. IEEE, 2009.

[5] Fangruo Chen and Yu-Sheng Zheng. Lower bounds for multi-echelon stochastic inventory systems. *Management Science*, 40(11):1426–1443, 1994.

[6] Leon Yang Chu, J George Shanthikumar, and Zuo-Jun Max Shen. Solving operational statistics via a bayesian analysis. *Operations Research Letters*, 36(1):110–116, 2008.

[7] Andrew J Clark and Herbert Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.

[8] Awi Federgruen and Paul Zipkin. Computational issues in an infinite-horizon, multiechelon inventory model. *Operations Research*, 32(4):818–836, 1984.

[9] Brant E. Fries. Optimal ordering policy for a perishable commodity with fixed lifetime. *Operations Research*, 23(1):46–61, 1975.

[10] M.C. Fu. *Handbook in OR & MS Vol. 13*, chapter Gradient Estimation.

[11] M.C. Fu and K.J. Healy. Techniques for optimization via simulation: an experimental study on an (s, s) inventory system. *IIE transactions*, 29(3):191–199, 1997.

[12] Guillermo Gallego and Ilkyeong Moon. The distribution free newsboy problem: Review and extensions. *Journal of the Operational Research Society*, 44(8):825–834, Aug 1993.

[13] Gregory A Godfrey and Warren B Powell. An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Science*, 47(8):1101–1112, 2001.

[14] S.K. Goyal and B.C. Giri. Recent trends in modeling of deteriorating inventory. *European Journal of Operational Research*, 134(1):1 – 16, 2001.

[15] Stephen C Graves et al. *Safety stocks in manufacturing systems*. Sloan School of Management, Massachusetts Institute of Technology, 1987.

[16] Stephen C Graves and Sean P Willems. Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, 2(1):68–83, 2000.

[17] Kevin Healy and Lee W Schruben. Retrospective simulation response optimization. In *Proceedings of the 23rd conference on Winter simulation*, pages 901–906. IEEE Computer Society, 1991.

[18] Woonghee Tim Huh and Paat Rusmevichientong. A nonparametric asymptotic analysis of inventory planning with censored demand. *Mathematics of Operations Research*, 34(1):103–123, 2009.

[19] Karl Inderfurth. Safety stock optimization in multi-stage inventory systems. *International Journal of Production Economics*, 24(1):103–113, 1991.

[20] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[21] Sujin Kim, Raghu Pasupathy, and Shane G Henderson. A guide to sample-average approximation. 2011.

[22] Retsef Levi, Robin O Roundy, and David B Shmoys. Provably near-optimal sampling-based policies for stochastic inventory control models. *Mathematics of Operations Research*, 32(4):821–839, 2007.

[23] Thomas L Magnanti, Zuo-Jun Max Shen, Jia Shu, David Simchi-Levi, and Chung-Piaw Teo. Inventory placement in acyclic supply chain networks. *Operations Research Letters*, 34(2):228–238, 2006.

[24] Wai-Kei Mak, David P Morton, and R Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24(1):47–56, 1999.

[25] Steven Nahmias. Optimal ordering policies for perishable inventory-ii. *Operations Research*, 23(4):735–749, 1975.

[26] Steven Nahmias. Perishable inventory theory: A review. *Operations Research*, 30(4):680–708, 1982.

[27] Raghu Pasupathy. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research*, 58(4-Part-1):889–901, 2010.

[28] Vivek Ramamurthy, J George Shanthikumar, and Zuo-Jun Max Shen. Inventory policy with parametric demand: Operational statistics, linear correction, and regression. *Production and Operations Management*, 21(2):291–308, 2012.

[29] S.M. Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21(3):513–528, 1996.

[30] J.O. Royset and R. Szechtman. Optimal budget allocation for sample average approximation. Technical report, DTIC Document, 2011.

[31] Herbert Scarf, KJ Arrow, and S Karlin. A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*, 10(2):201, 1958.

[32] Kevin H Shang and Jing-Sheng Song. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Management Science*, 49(5):618–638, 2003.

[33] Kenneth F Simpson. In-process inventories. *Operations Research*, 6(6):863–873, 1958.

[34] Shing Chih Tsai and Ya-Xin Zheng. A simulation optimization approach for a two-echelon inventory system with service level constraints. *European Journal of Operational Research*, 229(2):364–374, 2013.

[35] G. J. J. Van Zyl. Inventory control for perishable commodities. *Ph.D. dissertation*, 1964. Columbia University.

[36] Bram Verweij, Shabbir Ahmed, Anton J Kleywegt, George Nemhauser, and Alexander Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24(2):289–333, 2003.

[37] Huifu Xu and Dali Zhang. Smooth sample average approximation of stationary points in nonsmooth stochastic optimization and applications. *Mathematical programming*, 119(2):371–401, 2009.

[38] Paul Zipkin. Critical number policies for inventory models with periodic data. *Management Science*, 35(1):71–80, 1989.

Appendix A

Computational Results for Serial Supply Chain under Various Demand Distributions

This appendix displays parts of the comprehensive experimental results. With each distribution, we run a total of 100 sets of independent experiments. We include 10 of those 100 sets here.

A.1 Normal(32, 4)

Table A.1: Normal(32, 4): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					939390.8378	4696.954189	N/A
GS-200	16.07598726	0	0	0	833910.8878	4169.554439	1739
400-Period					1910605.953	4776.514881	N/A
GS-400	16.04234378	0	0	0	1677968.846	4194.922115	1710
1000-Period					4825669.511	4825.669511	N/A
GS-1000	16.02235537	0	0	0	4212844.447	4212.844447	1693
ROIP-5	37.71799846	34.63720037	33.34361012	22.46616117			
200-Period					65160.48498	325.8024249	N/A
GS-200	39.71799846	36.30386704	34.86265774	29.70088387	48495.80655	242.4790327	280
400-Period					133831.8006	334.5795014	N/A
GS-400	39.71799846	35.80386704	35.52932441	29.59760272	97044.07217	242.6101804	289
1000-Period					343302.8188	343.3028188	N/A
GS-1000	39.80133179	35.63720037	35.52932441	29.71530113	242471.2166	242.4712166	294
ROIP-10	37.71799846	34.63720037	33.34361012	31.74053226			
200-Period					49935.40625	249.6770312	N/A
GS-200	39.71799846	35.63720037	33.34361012	31.74053226	48362.31061	241.8115531	3
400-Period					99664.09651	249.1602413	N/A
GS-400	39.71799846	35.63720037	33.34361012	31.74053226	96795.82076	241.9895519	3
1000-Period					251459.2497	251.4592497	N/A
GS-1000	39.71799846	35.30386704	33.34361012	32.74053226	241818.7804	241.8187804	4
ROIP-20	38.68968933	37.04748306	36.94818786	30.47874387			
200-Period					48612.42288	243.0621144	N/A
GS-200	38.68968933	35.04748306	36.94818786	30.47874387	48486.11297	242.4305648	2
400-Period					97579.51592	243.9487898	N/A
GS-400	38.68968933	35.04748306	36.94818786	30.47874387	97099.54983	242.7488746	2
1000-Period					245586.9171	245.5869171	N/A
GS-1000	38.68968933	35.04748306	36.94818786	30.47874387	242676.9603	242.6769603	2
ROIP-30	37.71799846	36.94818786	36.94818786	30.47874387			
200-Period					48720.86576	243.6043288	N/A
GS-200	37.71799846	36.94818786	36.94818786	30.47874387	48843.7623	244.2188115	1
400-Period					97654.63468	244.1365867	N/A
GS-400	37.71799846	36.94818786	36.94818786	30.47874387	97851.14961	244.627874	1
1000-Period					246140.18	246.14018	N/A
GS-1000	37.71799846	36.94818786	36.94818786	30.47874387	244535.1888	244.5351888	1
ROIP-50	41.02851544	34.60936175	36.47174101	30.47874387			
200-Period					48234.39427	241.1719714	N/A
GS-200	39.02851544	34.60936175	36.47174101	30.47874387	48422.34991	242.1117496	2
400-Period					97195.92985	242.9898246	N/A
GS-400	39.02851544	34.60936175	36.47174101	30.47874387	96934.966	242.337415	2
1000-Period					244385.8456	244.3858456	N/A
GS-1000	39.02851544	34.60936175	36.47174101	30.47874387	242237.3511	242.2373511	2

Table A.2: Normal(32, 4): Experiment Run 2

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					950100.4462	4750.502231	N/A
GS-200	16.04935108	0	0	0	833225.7246	4166.128623	1716
400-Period					1907832.01	4769.580026	N/A
GS-400	16.0458505	0	0	0	1678188.326	4195.470815	1713
1000-Period					4772161.361	4772.161361	N/A
GS-1000	16.0481849	0	0	0	4212718.392	4212.718392	1715
ROIP-5	34.58128179	28.33214247	26.92068109	26.79228966			
200-Period					125231.1756	626.1558779	N/A
GS-200	38.03263385	33.25259146	33.230735	28.4671709	52150.21565	260.7510783	2001
400-Period					245500.4063	613.7510157	N/A
GS-400	39.14794845	33.15614652	32.81971808	27.85931816	104653.4291	261.6335728	2001
1000-Period					603684.6622	603.6846622	N/A
GS-1000	39.38278322	34.38935047	32.26423772	26.94675981	262218.1236	262.2181236	2001
ROIP-10	36.57034105	32.99499492	30.08709376	32.99499492			
200-Period					54860.32566	274.3016283	N/A
GS-200	39.57034105	33.41166159	34.04810103	33.32616042	48045.06815	240.2253407	27
400-Period					108526.8305	271.3170763	N/A
GS-400	39.23700772	34.11399259	33.59214684	32.99499492	96497.09754	241.2427438	22
1000-Period					275088.3455	275.0883455	N/A
GS-1000	39.57034105	34.19737588	32.76381785	32.99499492	241723.0457	241.7230457	18
ROIP-20	39.42904475	37.22428249	34.59786645	36.87912118			
200-Period					49083.04008	245.4152004	N/A
GS-200	37.54669181	35.05761582	34.19786645	34.57321875	48423.06041	242.1153021	19
400-Period					98643.02321	246.607558	N/A
GS-400	38.09467727	34.80761582	33.86453312	34.51715725	96879.57192	242.1989298	34
1000-Period					248628.1411	248.6281411	N/A
GS-1000	38.39987471	35.4850668	32.75641654	34.00583096	241992.6157	241.9926157	63
ROIP-30	37.22428249	37.22428249	36.80262871	36.87912118			
200-Period					49344.71708	246.7235854	N/A
GS-200	37.6885682	34.55761582	34.56929538	34.61149054	48427.21689	242.1360845	26
400-Period					99090.17147	247.7254287	N/A
GS-400	38.17465137	34.22428249	34.37802554	34.60060763	96964.52546	242.4113137	43
1000-Period					250763.4349	250.7634349	N/A
GS-1000	38.60731013	35.22428249	32.81723044	33.85299008	241838.6667	241.8386667	102
ROIP-50	40.84962133	31.9694242	36.29182816	37.22428249			
200-Period					48320.40854	241.6020427	N/A
GS-200	39.09962133	32.54953298	35.16682816	34.18917511	48305.57573	241.5278786	19
400-Period					96927.21686	242.3180422	N/A
GS-400	38.97462133	33.00587964	34.89182816	34.15382158	96874.40026	242.1860006	26
1000-Period					244453.8946	244.4538946	N/A
GS-1000	40.84962133	31.9694242	36.29182816	37.22428249	246150.1224	246.1501224	1

Table A.3: Normal(32, 4): Experiment Run 3

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					950374.7062	4751.873531	N/A
GS-200	16.04935108	0	0	0	833623.7861	4168.118931	1716
400-Period					1918135.743	4795.339356	N/A
GS-400	16.03413743	0	0	0	1678540.745	4196.351862	1703
1000-Period					4829152.846	4829.152846	N/A
GS-1000	16.02117334	0	0	0	4211143.512	4211.143512	1692
ROIP-5	36.45522299	33.23629849	32.33587056	30.31273088			
200-Period					53565.49605	267.8274803	N/A
GS-200	38.45522299	35.40296516	33.11768874	32.91616239	48247.73178	241.2386589	25
400-Period					110955.3513	277.3883783	N/A
GS-400	40.12188966	35.00534611	32.47244463	32.91803897	96404.8097	241.0120242	34
1000-Period					288884.0292	288.8840292	N/A
GS-1000	40.62188966	34.53882189	32.59616674	34.24891815	242005.3645	242.0053645	71
ROIP-10	36.45522299	33.23629849	32.33587056	30.31273088			
200-Period					53565.49605	267.8274803	N/A
GS-200	38.45522299	35.40296516	33.11768874	32.91616239	48247.73178	241.2386589	25
400-Period					110955.3513	277.3883783	N/A
GS-400	40.12188966	35.00534611	32.47244463	32.91803897	96404.8097	241.0120242	34
1000-Period					288884.0292	288.8840292	N/A
GS-1000	40.62188966	34.53882189	32.59616674	34.24891815	242005.3645	242.0053645	71
ROIP-20	39.36354509	35.89473398	32.35858171	25.12432568			
200-Period					51893.1347	259.4656735	N/A
GS-200	39.36354509	35.89473398	34.35858171	28.31003996	48526.40487	242.6320243	8
400-Period					105998.5011	264.9962528	N/A
GS-400	41.0673266	35.89473398	32.35858171	30.6110207	96792.19843	241.9804961	28
1000-Period					277720.1119	277.7201119	N/A
GS-1000	41.1225822	35.89473398	32.35858171	31.80095714	242704.2705	242.7042705	49
ROIP-30	39.36354509	35.89473398	32.35858171	31.26050535			
200-Period					48336.48367	241.6824184	N/A
GS-200	39.36354509	35.89473398	32.35858171	31.26050535	48281.30006	241.4065003	1
400-Period					96949.88394	242.3747098	N/A
GS-400	39.36354509	35.89473398	32.35858171	33.26050535	96445.83594	241.1145899	2
1000-Period					248215.8184	248.2158184	N/A
GS-1000	39.36354509	35.89473398	32.35858171	34.26050535	241891.2124	241.8912124	3
ROIP-50	39.76142657	35.4968525	33.96526024	30.82847565			
200-Period					48347.64247	241.7382124	N/A
GS-200	39.76142657	35.4968525	33.96526024	30.82847565	48224.35289	241.1217644	1
400-Period					96591.11162	241.477779	N/A
GS-400	39.76142657	35.4968525	33.96526024	30.82847565	96516.66643	241.2916661	1
1000-Period					246544.1162	246.5441162	N/A
GS-1000	39.76142657	35.4968525	33.96526024	32.82847565	242273.9152	242.2739152	2

Table A.4: Normal(32, 4): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					945896.6213	4729.483107	N/A
GS-200	16.05981622	0	0	0	833003.7723	4165.018861	1725
400-Period					1904462.605	4761.156513	N/A
GS-400	16.05051658	0	0	0	1676598.999	4191.497497	1717
1000-Period					4781613.026	4781.613026	N/A
GS-1000	16.04351337	0	0	0	4213107.016	4213.107016	1711
ROIP-5	37.13738344	35.54705326	28.32277346	31.76762456			
200-Period					52823.84559	264.119228	N/A
GS-200	39.13738344	35.54705326	31.75848775	31.76762456	48266.28606	241.3314303	9
400-Period					106466.2666	266.1656664	N/A
GS-400	39.63738344	35.54705326	31.25848775	31.76762456	96831.8098	242.0795245	9
1000-Period					262875.1686	262.8751686	N/A
GS-1000	39.13738344	35.54705326	31.75848775	32.37166497	242164.8133	242.1648133	12
ROIP-10	37.13738344	36.5785735	35.54705326	32.05467687			
200-Period					47967.48205	239.8374102	N/A
GS-200	38.13738344	34.5785735	34.88038659	32.05467687	48230.64602	241.1532301	4
400-Period					96582.02103	241.4550526	N/A
GS-400	38.13738344	34.5785735	35.54705326	32.05467687	96824.48505	242.0612126	3
1000-Period					242205.0399	242.2050399	N/A
GS-1000	38.13738344	34.5785735	34.38038659	32.05467687	242728.1607	242.7281607	5
ROIP-20	39.18096799	34.61998175	35.46206046	35.22312822			
200-Period					48123.406	240.61703	N/A
GS-200	39.18096799	34.61998175	35.46206046	35.22312822	48561.83638	242.8091819	1
400-Period					96677.85409	241.6946352	N/A
GS-400	40.28941772	32.61998175	33.79539379	32.36632827	96598.14779	241.4953695	26
1000-Period					242573.831	242.573831	N/A
GS-1000	39.18096799	34.61998175	35.46206046	35.22312822	244002.8011	244.0028011	1
ROIP-30	39.09597519	34.61998175	35.54705326	32.58736372			
200-Period					47490.57271	237.4528635	N/A
GS-200	39.09597519	34.61998175	35.54705326	32.58736372	48167.73096	240.8386548	1
400-Period					95411.02615	238.5275654	N/A
GS-400	39.76264186	32.61998175	34.54705326	32.58736372	96626.31385	241.5657846	4
1000-Period					239925.625	239.925625	N/A
GS-1000	39.09597519	34.61998175	35.54705326	32.58736372	242114.9158	242.1149158	1
ROIP-50	39.18096799	35.67941389	34.40262832	32.05467687			
200-Period					47435.46128	237.1773064	N/A
GS-200	39.18096799	35.67941389	34.40262832	32.05467687	48121.51048	240.6075524	1
400-Period					95384.29748	238.4607437	N/A
GS-400	39.18096799	33.67941389	34.40262832	32.05467687	96577.56833	241.4439208	2
1000-Period					239692.7063	239.6927063	N/A
GS-1000	39.18096799	35.67941389	34.40262832	32.05467687	241869.5973	241.8695973	1

Table A.5: Normal(32, 4): Experiment Run 5

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					951099.3146	4755.496573	N/A
GS-200	16.04701804	0	0	0	833297.28	4166.4864	1714
400-Period					1921596.877	4803.992194	N/A
GS-400	16.02943293	0	0	0	1677375.259	4193.438148	1699
1000-Period					4811004.349	4811.004349	N/A
GS-1000	16.02943293	0	0	0	4212419.028	4212.419028	1699
ROIP-5	40.57789076	30.55906545	33.47700698	23.92223184			
200-Period					60735.26777	303.6763389	N/A
GS-200	40.57789076	31.84477973	35.47700698	29.78737834	48617.59124	243.0879562	55
400-Period					126405.649	316.0141226	N/A
GS-400	40.57789076	32.67519809	36.26263392	28.87547333	96803.1669	242.0079172	78
1000-Period					326561.6255	326.5616255	N/A
GS-1000	40.57789076	34.29472978	36.37780063	28.94998307	241982.3136	241.9823136	192
ROIP-10	40.57789076	30.55906545	33.47700698	23.92223184			
200-Period					60735.26777	303.6763389	N/A
GS-200	40.57789076	31.84477973	35.47700698	29.78737834	48617.59124	243.0879562	55
400-Period					126405.649	316.0141226	N/A
GS-400	40.57789076	32.67519809	36.26263392	28.87547333	96803.1669	242.0079172	78
1000-Period					326561.6255	326.5616255	N/A
GS-1000	40.57789076	34.29472978	36.37780063	28.94998307	241982.3136	241.9823136	192
ROIP-20	40.57789076	30.55906545	34.64238637	33.93087902			
200-Period					47433.16385	237.1658192	N/A
GS-200	40.57789076	31.55906545	34.64238637	31.93087902	48437.24801	242.1862401	3
400-Period					98950.82055	247.3770514	N/A
GS-400	40.57789076	32.55906545	34.64238637	33.93087902	96565.56239	241.413906	2
1000-Period					251109.8559	251.1098559	N/A
GS-1000	40.57789076	32.55906545	34.64238637	33.93087902	241876.7929	241.8767929	2
ROIP-30	40.57789076	32.42720474	33.74801085	31.84561375			
200-Period					47132.90526	235.6645263	N/A
GS-200	40.57789076	32.42720474	33.74801085	31.84561375	48360.56759	241.8028379	1
400-Period					98362.87678	245.9071919	N/A
GS-400	40.57789076	32.42720474	33.74801085	31.84561375	96710.562	241.776405	1
1000-Period					249012.5868	249.0125868	N/A
GS-1000	40.57789076	34.42720474	33.74801085	31.84561375	241514.6471	241.5146471	2
ROIP-50	40.57789076	30.55906545	35.69436342	31.76740048			
200-Period					47314.6942	236.573471	N/A
GS-200	40.57789076	30.55906545	35.69436342	31.76740048	48680.25868	243.4012934	1
400-Period					98848.24649	247.1206162	N/A
GS-400	40.57789076	32.55906545	35.69436342	31.76740048	96485.21985	241.2130496	2
1000-Period					251102.7454	251.1027454	N/A
GS-1000	40.57789076	33.55906545	35.69436342	31.76740048	242062.9013	242.0629013	3

Table A.6: Normal(32, 4): Experiment Run 6

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					941640.4198	4708.202099	N/A
GS-200	16.07022689	0	0	0	832111.6936	4160.558468	1734
400-Period					1892872.261	4732.180653	N/A
GS-400	16.06444988	0	0	0	1675667.624	4189.16906	1729
1000-Period					4778597.467	4778.597467	N/A
GS-1000	16.04468227	0	0	0	4208638.428	4208.638428	1712
ROIP-5	36.5577253	31.03863701	27.27193904	27.27416612			
200-Period					81968.12731	409.8406366	N/A
GS-200	43.29993296	31.49226659	31.08424742	30.90267235	48976.24639	244.8812319	847
400-Period					166152.3602	415.3809005	N/A
GS-400	40.87439196	33.89240292	31.30081413	30.7115103	97840.6555	244.6016388	847
1000-Period					441274.4401	441.2744401	N/A
GS-1000	40.22439196	34.38895979	32.02084417	30.74989595	243575.3277	243.5753277	1146
ROIP-10	43.5961189	31.19839602	29.60123523	28.60496284			
200-Period					48976.65112	244.8832556	N/A
GS-200	43.5961189	31.19839602	31.60123523	30.27162951	49015.34732	245.0767366	4
400-Period					99256.70112	248.1417528	N/A
GS-400	43.5961189	31.19839602	31.60123523	30.77162951	97769.66676	244.4241669	5
1000-Period					259438.3217	259.4383217	N/A
GS-1000	43.5961189	31.19839602	31.60123523	32.46289935	242727.0545	242.7270545	11
ROIP-20	43.5961189	31.3656449	38.61084981	33.68997591			
200-Period					49783.62094	248.9181047	N/A
GS-200	40.5961189	32.04236408	34.29704107	31.98666129	48322.17745	241.6108872	72
400-Period					100182.0984	250.4552461	N/A
GS-400	40.5961189	32.18478081	33.43392569	32.1766157	96924.99184	242.3124796	108
1000-Period					250850.035	250.850035	N/A
GS-1000	40.5961189	31.3656449	34.97439182	33.68997591	242253.2667	242.2532667	16
ROIP-30	43.5961189	31.3656449	38.61084981	31.19839602			
200-Period					49117.40588	245.5870294	N/A
GS-200	40.5961189	31.3656449	35.40442845	31.19839602	48457.74412	242.2887206	13
400-Period					98864.58983	247.1614746	N/A
GS-400	40.5961189	31.3656449	35.10772516	31.19839602	97134.51556	242.8362889	15
1000-Period					247715.3121	247.7153121	N/A
GS-1000	40.5961189	31.96949106	35.85442845	31.19839602	241907.6681	241.9076681	14
ROIP-50	43.5961189	31.20655922	31.19839602	31.19839602			
200-Period					47058.76206	235.2938103	N/A
GS-200	43.5961189	31.20655922	31.19839602	31.19839602	48830.40943	244.1520472	1
400-Period					95005.59893	237.5139973	N/A
GS-400	43.5961189	31.20655922	31.19839602	31.19839602	97776.55657	244.4413914	1
1000-Period					244424.7812	244.4247812	N/A
GS-1000	43.5961189	31.20655922	31.19839602	33.19839602	242665.9271	242.6659271	2

Table A.7: Normal(32, 4): Experiment Run 7

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					950003.631	4750.018155	N/A
GS-200	16.04935108	0	0	0	832982.935	4164.914675	1716
400-Period					1900414.705	4751.036762	N/A
GS-400	16.05517181	0	0	0	1678068.882	4195.172204	1721
1000-Period					4768882.009	4768.882009	N/A
GS-1000	16.04935108	0	0	0	4210324.88	4210.32488	1716
ROIP-5	38.82826203	30.54814665	31.71033534	34.86971551			
200-Period					52925.04703	264.6252352	N/A
GS-200	40.82826203	33.73386094	32.54922423	34.68789733	48131.30024	240.6565012	13
400-Period					101923.008	254.8075199	N/A
GS-400	40.82826203	33.04814665	31.71033534	34.18400123	96854.47582	242.1361896	8
1000-Period					257793.9356	257.7939356	N/A
GS-1000	40.82826203	33.29814665	31.71033534	33.96177901	242079.5823	242.0795823	10
ROIP-10	38.82826203	33.81419085	34.48564644	34.80012163			
200-Period					48923.92116	244.6196058	N/A
GS-200	38.82826203	33.81419085	34.48564644	34.80012163	48249.13764	241.2456882	1
400-Period					97077.16329	242.6929082	N/A
GS-400	38.82826203	33.81419085	34.48564644	34.80012163	96957.67457	242.3941864	1
1000-Period					242520.8331	242.5208331	N/A
GS-1000	39.82826203	33.81419085	32.48564644	33.63345496	241855.5534	241.8555534	5
ROIP-20	38.82826203	35.06838203	35.58028249	34.51928892			
200-Period					48925.17917	244.6258958	N/A
GS-200	38.82826203	35.06838203	35.58028249	34.51928892	48548.36341	242.7418171	1
400-Period					97764.51727	244.4112932	N/A
GS-400	38.82826203	35.06838203	35.58028249	34.51928892	97488.78754	243.7219688	1
1000-Period					244177.3441	244.1773441	N/A
GS-1000	38.82826203	35.06838203	35.58028249	34.51928892	243995.289	243.995289	1
ROIP-30	38.82826203	35.06838203	35.23135202	34.51928892			
200-Period					48872.47068	244.3623534	N/A
GS-200	38.82826203	35.06838203	35.23135202	34.51928892	48460.88529	242.3044265	1
400-Period					97594.19903	243.9854976	N/A
GS-400	38.82826203	35.06838203	35.23135202	34.51928892	97321.38921	243.303473	1
1000-Period					243704.2373	243.7042373	N/A
GS-1000	38.82826203	35.06838203	35.23135202	34.51928892	243566.8941	243.5668941	1
ROIP-50	40.76909037	39.71990386	34.52980705	28.87767149			
200-Period					49612.35494	248.0617747	N/A
GS-200	38.76909037	39.71990386	34.52980705	28.87767149	48730.96716	243.6548358	2
400-Period					99385.53165	248.4638291	N/A
GS-400	37.76909037	37.28418957	34.52980705	28.87767149	97457.43105	243.6435776	9
1000-Period					248679.8755	248.6798755	N/A
GS-1000	38.76909037	36.53418957	34.52980705	28.87767149	242664.8441	242.6648441	8

Table A.8: Normal(32, 4): Experiment Run 8

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					963145.0698	4815.725349	N/A
GS-200	16.01762304	0	0	0	833694.3364	4168.471682	1689
400-Period					1926930.671	4817.326678	N/A
GS-400	16.0235367	0	0	0	1676650.272	4191.625681	1694
1000-Period					4791392.876	4791.392876	N/A
GS-1000	16.0388309	0	0	0	4207205.212	4207.205212	1707
ROIP-5	39.01643217	28.87353513	38.49672014	26.35764703			
200-Period					62303.82821	311.519141	N/A
GS-200	41.01643217	34.18494851	39.09546619	27.80722842	48585.58926	242.9279463	61
400-Period					114560.2955	286.4007387	N/A
GS-400	41.01643217	32.53147164	38.49672014	28.25162687	96790.11233	241.9752808	25
1000-Period					274178.9496	274.1789496	N/A
GS-1000	41.01643217	32.53147164	38.49672014	27.20283517	242305.7429	242.3057429	15
ROIP-10	40.92890345	32.58314104	32.87464294	27.4750774			
200-Period					57324.06905	286.6203453	N/A
GS-200	40.92890345	34.58314104	37.6958	28.70841073	48529.79159	242.6489579	32
400-Period					106181.5822	265.4539555	N/A
GS-400	40.92890345	34.58314104	34.54130961	29.84816542	96654.74892	241.6368723	12
1000-Period					257497.469	257.497469	N/A
GS-1000	40.92890345	34.58314104	34.37464294	29.16079168	242179.2536	242.1792536	8
ROIP-20	42.04067355	34.36542938	38.07746094	29.11308771			
200-Period					49187.53322	245.9376661	N/A
GS-200	41.04067355	34.36542938	38.07746094	29.11308771	48667.62595	243.3381298	3
400-Period					97881.25026	244.7031256	N/A
GS-400	40.04067355	34.36542938	38.07746094	29.11308771	96962.92127	242.4073032	2
1000-Period					246027.7494	246.0277494	N/A
GS-1000	39.04067355	34.36542938	37.41079427	29.11308771	242070.7201	242.0707201	4
ROIP-30	42.04067355	34.36542938	38.07746094	29.11308771			
200-Period					49187.53322	245.9376661	N/A
GS-200	41.04067355	34.36542938	38.07746094	29.11308771	48667.62595	243.3381298	3
400-Period					97881.25026	244.7031256	N/A
GS-400	40.04067355	34.36542938	38.07746094	29.11308771	96962.92127	242.4073032	2
1000-Period					246027.7494	246.0277494	N/A
GS-1000	39.04067355	34.36542938	37.41079427	29.11308771	242070.7201	242.0707201	4
ROIP-50	42.04067355	33.51967883	40.50889707	27.85803992			
200-Period					49213.44226	246.0672113	N/A
GS-200	40.04067355	33.51967883	40.50889707	27.85803992	48497.31173	242.4865587	2
400-Period					97929.12218	244.8228054	N/A
GS-400	40.04067355	33.51967883	40.50889707	27.85803992	97225.02686	243.0625672	2
1000-Period					246266.2183	246.2662183	N/A
GS-1000	39.04067355	33.51967883	39.34223041	27.85803992	242622.3748	242.6223748	5

Table A.9: Normal(32, 4): Experiment Run 9

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					940051.0932	4700.255466	N/A
GS-200	16.0736851	0	0	0	833364.2522	4166.821261	1737
400-Period					1909377.231	4773.443077	N/A
GS-400	16.04468227	0	0	0	1677832.74	4194.581851	1712
1000-Period					4793339.519	4793.339519	N/A
GS-1000	16.03765856	0	0	0	4209364.623	4209.364623	1706
ROIP-5	31.89015871	26.35138447	26.95941756	30.04024744			
200-Period					127893.1502	639.4657508	N/A
GS-200	42.25283841	31.4385328	27.86632574	30.04024744	55029.07113	275.1453557	2001
400-Period					274303.7922	685.7594805	N/A
GS-400	42.18122662	31.41105918	27.96541115	30.04024744	110371.0197	275.9275493	2001
1000-Period					690048.584	690.048584	N/A
GS-1000	41.3159635	33.28231589	26.95941756	30.04024744	275865.8547	275.8658547	2001
ROIP-10	32.68316114	31.8386233	31.89015871	28.0321774			
200-Period					72017.0875	360.0854375	N/A
GS-200	40.6731354	34.04634241	31.89015871	29.48822205	49198.53953	245.9926976	191
400-Period					160506.5024	401.266256	N/A
GS-400	39.97387855	33.50631657	33.95146757	31.48931963	96656.86117	241.6421529	782
1000-Period					405062.0436	405.0620436	N/A
GS-1000	40.2350775	34.86605968	32.53735093	31.20686592	242040.3351	242.0403351	753
ROIP-20	42.04436159	30.42712665	26.08864256	28.70058404			
200-Period					59420.00361	297.100018	N/A
GS-200	42.04436159	33.76395112	30.6391327	29.46675504	49405.75029	247.0287514	43
400-Period					133812.4293	334.5310733	N/A
GS-400	42.04436159	31.21284094	34.37836076	32.39286336	96653.01267	241.6325317	333
1000-Period					337603.8976	337.6038976	N/A
GS-1000	42.04436159	33.42712665	32.12197579	32.57435127	242025.5798	242.0255798	357
ROIP-30	42.35135963	37.69167366	30.42712665	23.37379202			
200-Period					47182.6186	235.913093	N/A
GS-200	42.35135963	37.69167366	30.42712665	25.37379202	49607.86151	248.0393075	2
400-Period					102174.5757	255.4364391	N/A
GS-400	42.35135963	37.69167366	30.42712665	27.94045869	97626.21877	244.0655469	6
1000-Period					259021.8835	259.0218835	N/A
GS-1000	42.35135963	37.69167366	30.42712665	28.27379202	244234.9348	244.2349348	7
ROIP-50	42.35135963	37.69167366	30.42712665	26.06203046			
200-Period					46454.02092	232.2701046	N/A
GS-200	42.35135963	37.69167366	30.42712665	26.06203046	49238.72593	246.1936297	1
400-Period					98086.98906	245.2174726	N/A
GS-400	42.35135963	37.69167366	30.42712665	28.06203046	97591.55501	243.9788875	2
1000-Period					248681.5896	248.6815896	N/A
GS-1000	42.35135963	37.69167366	30.42712665	28.06203046	244358.2849	244.3582849	2

Table A.10: Normal(32, 4): Experiment Run 10

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
200-Period					951362.1143	4756.810571	N/A
GS-200	16.04701804	0	0	0	831674.1957	4158.370978	1714
400-Period					1909298.764	4773.24691	N/A
GS-400	16.04468227	0	0	0	1676479.102	4191.197754	1712
1000-Period					4794532.454	4794.532454	N/A
GS-1000	16.03765856	0	0	0	4209384.477	4209.384477	1706
ROIP-5	38.77038767	32.13737569	31.00822025	30.83975668			
200-Period					58346.60162	291.7330081	N/A
GS-200	40.56537091	37.43618521	32.1651113	32.41809992	48528.16741	242.6408371	77
400-Period					109993.4276	274.983569	N/A
GS-400	39.77038767	36.08499474	33.09187279	31.92547548	96509.56679	241.273917	33
1000-Period					271973.8535	271.9738535	N/A
GS-1000	40.77038767	33.80404235	33.49755773	32.39259197	241455.2066	241.4552066	27
ROIP-10	38.77038767	32.13737569	31.00822025	30.83975668			
200-Period					58346.60162	291.7330081	N/A
GS-200	40.56537091	37.43618521	32.1651113	32.41809992	48528.16741	242.6408371	77
400-Period					109993.4276	274.983569	N/A
GS-400	39.77038767	36.08499474	33.09187279	31.92547548	96509.56679	241.273917	33
1000-Period					271973.8535	271.9738535	N/A
GS-1000	40.77038767	33.80404235	33.49755773	32.39259197	241455.2066	241.4552066	27
ROIP-20	38.77038767	32.13737569	31.00822025	33.04327801			
200-Period					55750.89451	278.7544726	N/A
GS-200	40.77038767	35.95042098	33.18514923	33.79875553	48627.74067	243.1387033	45
400-Period					105547.7904	263.8694761	N/A
GS-400	40.77038767	34.85959791	32.14393454	33.04327801	96442.84331	241.1071083	11
1000-Period					261240.4993	261.2404993	N/A
GS-1000	40.77038767	33.80404235	32.77726787	33.04327801	241424.2433	241.4242433	9
ROIP-30	38.77038767	33.37903338	29.76656256	30.98751923			
200-Period					57979.51832	289.8975916	N/A
GS-200	40.61921767	37.33001377	32.07867998	32.48370801	48498.80489	242.4940244	69
400-Period					109269.1366	273.1728414	N/A
GS-400	39.77038767	36.02903338	33.16118299	31.93287306	96507.32124	241.2683031	31
1000-Period					270190.6055	270.1906055	N/A
GS-1000	40.77038767	34.04570004	33.25985261	32.29614554	241475.5997	241.4755997	24
ROIP-50	38.77038767	34.48424989	35.70249318	36.29405768			
200-Period					50219.95077	251.0997539	N/A
GS-200	38.77038767	34.48424989	35.70249318	36.29405768	48738.51089	243.6925544	1
400-Period					99093.24551	247.7331138	N/A
GS-400	39.45610196	34.48424989	33.70249318	33.54405768	96364.24572	240.9106143	9
1000-Period					246728.939	246.728939	N/A
GS-1000	38.77038767	34.48424989	35.70249318	36.29405768	244890.9477	244.8909477	1

A.2 Normal(32, 8)

Table A.11: Normal(32, 8): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					941535.1721	4707.67586	N/A
GS-200	16.07022689	0	0	0	833134.5824	4165.672912	1734
400-Period					1891773.172	4729.432929	N/A
GS-400	16.06560661	0	0	0	1679357.241	4198.393103	1730
1000-Period					4768112.03	4768.11203	N/A
GS-1000	16.05051658	0	0	0	4214167.593	4214.167593	1717
ROIP-5	39.81370529	28.35461206	32.97929742	31.78863641			
200-Period					74718.62233	373.5931116	N/A
GS-200	45.29281605	34.94832418	34.09496422	33.15234696	58823.52571	294.1176286	812
400-Period					159936.6292	399.8415729	N/A
GS-400	46.77830896	36.32039589	32.97929742	31.79694812	117139.9744	292.8499361	985
1000-Period					420776.1715	420.7761715	N/A
GS-1000	46.70906711	37.04930202	32.97929742	31.78863641	293303.0865	293.3030865	1364
ROIP-10	39.81370529	35.23273911	31.31052649	26.57928029			
200-Period					69193.47498	345.9673749	N/A
GS-200	45.13800418	35.23273911	34.09714786	33.02056026	58803.85166	294.0192583	812
400-Period					149024.4957	372.5612394	N/A
GS-400	46.43789521	38.20401612	32.78076574	30.23125555	117130.0315	292.8250788	882
1000-Period					393393.1528	393.3931528	N/A
GS-1000	46.77260596	37.81772729	35.10724498	28.66686613	293181.6702	293.1816702	1258
ROIP-20	44.62938316	30.41706124	32.29163158	33.94858615			
200-Period					60024.06949	300.1203474	N/A
GS-200	45.29604983	35.63277255	34.20815851	35.54220894	58150.95954	290.7547977	62
400-Period					129105.4206	322.7635516	N/A
GS-400	46.72495191	37.80786163	32.75795403	35.8205905	115880.3789	289.7009472	208
1000-Period					334417.4189	334.4174189	N/A
GS-1000	46.93824363	37.66208192	35.09272709	33.94858615	290683.2383	290.6832383	271
ROIP-30	43.64827807	31.39816633	31.31052649	33.94858615			
200-Period					61023.07384	305.1153692	N/A
GS-200	45.64827807	35.15432018	34.2083998	35.45970019	58165.81242	290.8290621	91
400-Period					131309.2898	328.2732244	N/A
GS-400	46.66570801	37.83303594	32.79602051	35.07158959	115871.8829	289.6797073	234
1000-Period					341074.76	341.07476	N/A
GS-1000	46.93472149	37.69503899	34.4088875	33.94858615	290612.6696	290.6126696	319
ROIP-50	47.00851139	34.21341751	35.6220065	35.51543111			
200-Period					55498.06691	277.4903346	N/A
GS-200	45.00851139	34.21341751	35.6220065	35.51543111	58526.05662	292.6302831	2
400-Period					116386.279	290.9656976	N/A
GS-400	47.00851139	36.88008418	34.6220065	35.51543111	115858.8993	289.6472482	4
1000-Period					294892.6106	294.8926106	N/A
GS-1000	47.00851139	36.21341751	35.6220065	35.51543111	290925.3617	290.9253617	2

Table A.12: Normal(32, 8): Experiment Run 2

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					943688.2359	4718.441179	N/A
GS-200	16.06560661	0	0	0	834679.3648	4173.396824	1730
400-Period					1881005.989	4702.514973	N/A
GS-400	16.07943554	0	0	0	1677047.698	4192.619245	1742
1000-Period					4767481.278	4767.481278	N/A
GS-1000	16.05051658	0	0	0	4212912.076	4212.912076	1717
ROIP-5	47.77915154	42.29878046	43.84995464	34.96323135			
200-Period					62294.39851	311.4719926	N/A
GS-200	45.77915154	39.50989157	35.03880325	34.67075115	58087.50039	290.4375019	584
400-Period					125298.6993	313.2467483	N/A
GS-400	45.77915154	38.3690278	35.79017785	34.96323135	116404.3604	291.0109011	613
1000-Period					314810.2368	314.8102368	N/A
GS-1000	45.77915154	37.64442867	37.56935038	34.96323135	292313.8117	292.3138117	362
ROIP-10	47.77915154	42.29878046	43.84995464	34.96323135			
200-Period					62294.39851	311.4719926	N/A
GS-200	45.77915154	39.50989157	35.03880325	34.67075115	58087.50039	290.4375019	584
400-Period					125298.6993	313.2467483	N/A
GS-400	45.77915154	38.3690278	35.79017785	34.96323135	116404.3604	291.0109011	613
1000-Period					314810.2368	314.8102368	N/A
GS-1000	45.77915154	37.64442867	37.56935038	34.96323135	292313.8117	292.3138117	362
ROIP-20	43.84995464	42.29878046	41.2616859	34.96323135			
200-Period					60125.67256	300.6283628	N/A
GS-200	45.81013047	40.29878046	34.72433121	32.73560483	58044.04401	290.22022	326
400-Period					120174.5138	300.4362844	N/A
GS-400	45.00766091	39.49479252	33.20919709	34.61505762	116394.9166	290.9872914	272
1000-Period					304960.2652	304.9602652	N/A
GS-1000	45.48856598	37.55666004	36.88815215	34.96323135	291881.5007	291.8815007	122
ROIP-30	43.84995464	42.29878046	41.2616859	34.96323135			
200-Period					60125.67256	300.6283628	N/A
GS-200	45.81013047	40.29878046	34.72433121	32.73560483	58044.04401	290.22022	326
400-Period					120174.5138	300.4362844	N/A
GS-400	45.00766091	39.49479252	33.20919709	34.61505762	116394.9166	290.9872914	272
1000-Period					304960.2652	304.9602652	N/A
GS-1000	45.48856598	37.55666004	36.88815215	34.96323135	291881.5007	291.8815007	122
ROIP-50	43.37725793	34.96323135	34.96323135	31.28367642			
200-Period					60881.69914	304.4084957	N/A
GS-200	48.78824694	36.45818008	34.96323135	32.23208075	57938.84083	289.6942042	29
400-Period					118811.9648	297.0299121	N/A
GS-400	48.37686111	35.4170775	34.96323135	32.70933213	116121.3408	290.303352	18
1000-Period					310781.2089	310.7812089	N/A
GS-1000	47.64918776	36.20226269	34.96323135	33.76268951	290792.6399	290.7926399	31

Table A.13: Normal(32, 8): Experiment Run 3

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
Experiment Run 3							
ROIP-0	0	0	0	0			
200-Period					959263.1901	4796.31595	N/A
GS-200	16.02707653	0	0	0	832405.5169	4162.027585	1697
400-Period					1910360.062	4775.900154	N/A
GS-400	16.04351337	0	0	0	1676025.994	4190.064986	1711
1000-Period					4802774.904	4802.774904	N/A
GS-1000	16.03296235	0	0	0	4213642.928	4213.642928	1702
ROIP-5	46.34991624	27.39088995	30.96237818	37.28212779			
200-Period					68160.47157	340.8023578	N/A
GS-200	46.84407313	36.94207614	32.61233296	37.28212779	57821.47511	289.1073756	195
400-Period					128239.64	320.5991	N/A
GS-400	46.34991624	36.50106311	31.20963999	36.20598042	117309.2963	293.2732407	104
1000-Period					334948.0571	334.9480571	N/A
GS-1000	46.34991624	37.00302461	32.97223491	37.28212779	292308.1885	292.3081885	188
ROIP-10	47.16203714	35.96339782	23.19349821	37.28212779			
200-Period					63982.61438	319.9130719	N/A
GS-200	47.16203714	36.6702462	32.56138113	37.28212779	57814.11061	289.070553	87
400-Period					121336.758	303.3418951	N/A
GS-400	47.16203714	35.96339782	31.22861886	35.82946664	117133.0813	292.8327033	65
1000-Period					318011.0657	318.0110657	N/A
GS-1000	47.16203714	36.14566287	33.06270864	37.28212779	292140.4296	292.1404296	86
ROIP-20	50.40247536	35.96339782	47.16203714	15.68714712			
200-Period					57103.44123	285.5172062	N/A
GS-200	49.40247536	35.96339782	47.16203714	18.35381379	58090.80358	290.4540179	4
400-Period					112046.7785	280.1169464	N/A
GS-400	48.40247536	35.96339782	47.16203714	16.68714712	117412.0959	293.5302396	3
1000-Period					291319.9825	291.3199825	N/A
GS-1000	50.40247536	35.96339782	47.16203714	15.68714712	294369.2967	294.3692967	1
ROIP-30	48.06655266	35.96339782	47.16203714	22.75787941			
200-Period					56461.97257	282.3098628	N/A
GS-200	48.06655266	35.96339782	47.16203714	22.75787941	58232.52114	291.1626057	1
400-Period					112560.1314	281.4003286	N/A
GS-400	45.06655266	35.96339782	45.99537048	22.75787941	117537.1898	293.8429745	5
1000-Period					290175.3215	290.1753215	N/A
GS-1000	46.06655266	35.96339782	46.16203714	22.75787941	293578.1068	293.5781068	3
ROIP-50	49.27955951	35.96339782	47.16203714	18.97895943			
200-Period					56605.73616	283.0286808	N/A
GS-200	49.27955951	35.96339782	47.16203714	18.97895943	58091.77182	290.4588591	1
400-Period					112006.7067	280.0167667	N/A
GS-400	47.27955951	35.96339782	47.16203714	18.97895943	117062.8653	292.6571633	2
1000-Period					290115.5003	290.1155003	N/A
GS-1000	47.27955951	35.96339782	47.16203714	19.97895943	293224.4435	293.2244435	3

Table A.14: Normal(32, 8): Experiment Run 4

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
Experiment Run 4							
ROIP-0	0	0	0	0			
200-Period					961879.2744	4809.396372	N/A
GS-200	16.02117334	0	0	0	834668.8212	4173.344106	1692
400-Period					1901002.955	4752.507387	N/A
GS-400	16.05400902	0	0	0	1680548.035	4201.370087	1720
1000-Period					4712696.323	4712.696323	N/A
GS-1000	16.07713735	0	0	0	4208185.529	4208.185529	1740
ROIP-5	36.19937804	28.47260658	29.95471552	26.23835119			
200-Period					132659.8004	663.2990019	N/A
GS-200	51.58011505	29.44860578	29.95471552	26.23835119	67840.025	339.200125	2001
400-Period					240673.7324	601.6843309	N/A
GS-400	49.93396789	31.09475294	29.95471552	26.23835119	136397.9142	340.9947855	2001
1000-Period					580362.1109	580.3621109	N/A
GS-1000	47.50711427	33.52160655	29.95471552	26.23835119	339142.2043	339.1422043	2001
ROIP-10	38.15589886	32.83068171	36.25405677	25.38756229			
200-Period					87197.74219	435.988711	N/A
GS-200	49.31141319	35.88348527	37.72472606	25.38756229	58731.32746	293.6566373	1426
400-Period					161210.2096	403.0255239	N/A
GS-400	48.32843866	35.11409065	36.25405677	28.13801028	117838.5138	294.5962846	1126
1000-Period					393292.9525	393.2929525	N/A
GS-1000	47.09932324	35.77838147	39.46351353	25.38756229	294436.8622	294.4368622	1068
ROIP-20	39.25170534	31.98671139	36.0022206	25.95102126			
200-Period					85332.7327	426.6636635	N/A
GS-200	49.29816993	35.98549061	37.28906029	25.95102126	58668.86727	293.3443364	1199
400-Period					158096.261	395.2406526	N/A
GS-400	48.94591732	34.73009545	36.01564342	28.42238332	117698.0771	294.2451928	977
1000-Period					385710.7592	385.7107592	N/A
GS-1000	47.12053857	35.76773952	39.1846726	25.95102126	294013.981	294.013981	934
ROIP-30	46.38698625	29.48413375	36.52712143	31.00163718			
200-Period					65654.44453	328.2722227	N/A
GS-200	49.38698625	34.63513577	36.52712143	33.53794919	58169.9706	290.849853	118
400-Period					126076.6446	315.1916114	N/A
GS-400	48.38698625	34.31874517	36.52712143	32.62282978	116562.6956	291.4067391	39
1000-Period					314180.7416	314.1807416	N/A
GS-1000	46.38698625	36.37237504	37.23854115	31.90906224	291357.5922	291.3575922	40
ROIP-50	46.38698625	42.78958375	26.82890806	44.05553774			
200-Period					58255.87962	291.2793981	N/A
GS-200	46.38698625	42.78958375	27.82890806	42.05553774	58931.14907	294.6557454	3
400-Period					117644.9753	294.1124382	N/A
GS-400	46.38698625	41.78958375	27.31608754	39.18244971	117405.7264	293.514316	14
1000-Period					297687.8591	297.6878591	N/A
GS-1000	46.38698625	42.78958375	28.68236096	35.23877063	291948.2777	291.9482777	117

Table A.15: Normal(32, 8): Experiment Run 5

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					957723.0427	4788.615213	N/A
GS-200	16.0306101	0	0	0	834487.1673	4172.435836	1700
400-Period					1926428.208	4816.07052	N/A
GS-400	16.0235367	0	0	0	1679624.86	4199.062151	1694
1000-Period					4805709.634	4805.709634	N/A
GS-1000	16.03178657	0	0	0	4209958.941	4209.958941	1701
ROIP-5	33.27225754	23.94118741	21.94570887	28.10019779			
200-Period					208786.9998	1043.934999	N/A
GS-200	45.3234272	28.24675396	21.94570887	28.10019779	109654.132	548.2706601	2001
400-Period					419517.6498	1048.794124	N/A
GS-400	45.28577383	28.28440733	21.94570887	28.10019779	219489.6987	548.7242467	2001
1000-Period					1031445.9	1031.4459	N/A
GS-1000	49.62899375	23.94118741	21.94570887	28.10019779	546373.1789	546.3731789	2001
ROIP-10	35.97692885	34.94657818	23.94118741	33.27225754			
200-Period					101781.0382	508.905191	N/A
GS-200	45.79260347	35.45531901	29.97350817	33.27225754	61272.2248	306.361124	2001
400-Period					205742.7984	514.356996	N/A
GS-400	46.91990217	40.31472744	23.98680104	33.27225754	122665.4076	306.6635191	2001
1000-Period					510817.656	510.817656	N/A
GS-1000	47.93548354	39.34475971	23.94118741	33.27225754	306072.2746	306.0722746	2001
ROIP-20	44.13602412	27.99562662	33.34719501	27.11968002			
200-Period					78224.01453	391.1200726	N/A
GS-200	48.58202018	32.77419805	35.55953817	30.88916609	58937.70204	294.6885102	1126
400-Period					160702.4462	401.7561155	N/A
GS-400	46.97606437	35.47350916	37.37437488	27.78902054	118068.4573	295.1711433	1023
1000-Period					399240.4121	399.2404121	N/A
GS-1000	47.81354372	36.81658154	33.94599545	28.96313474	295012.8617	295.0128617	986
ROIP-30	44.13602412	30.84557475	36.22955547	27.57452608			
200-Period					64893.91588	324.4695794	N/A
GS-200	50.19703392	30.84557475	36.83268362	33.99967595	58399.75284	291.9987642	391
400-Period					135057.1828	337.642957	N/A
GS-400	47.13602412	35.35006061	37.3691695	31.82062516	116680.464	291.7011599	354
1000-Period					336705.7088	336.7057088	N/A
GS-1000	47.63602412	36.09102654	36.22955547	31.19878397	291772.0066	291.7720066	273
ROIP-50	50.61736766	28.77477099	36.95070037	27.57452608			
200-Period					58294.0552	291.470276	N/A
GS-200	50.61736766	30.77477099	36.95070037	33.49783367	58357.7824	291.788912	30
400-Period					121368.081	303.4202024	N/A
GS-400	50.61736766	31.77477099	36.95070037	33.27452332	116783.7864	291.9594661	44
1000-Period					304584.0514	304.5840514	N/A
GS-1000	50.61736766	32.44143766	36.95070037	31.53977577	292188.3677	292.1883677	26

Table A.16: Normal(32, 8): Experiment Run 6

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
Experiment Run 6							
ROIP-0	0	0	0	0			
200-Period					931464.1754	4657.320877	N/A
GS-200	16.09544935	0	0	0	833590.5812	4167.952906	1756
400-Period					1895782.084	4739.455211	N/A
GS-400	16.06097564	0	0	0	1679012.802	4197.532006	1726
1000-Period					4751612.675	4751.612675	N/A
GS-1000	16.05865613	0	0	0	4209570.083	4209.570083	1724
ROIP-5	43.98153542	27.30970603	30.64688325	28.86102228			
200-Period					77767.68438	388.8384219	N/A
GS-200	48.30654808	33.61883439	33.09176045	30.7145995	59780.46454	298.9023227	982
400-Period					160318.6957	400.7967392	N/A
GS-400	48.98942198	33.58660688	32.66001877	30.31229522	120290.5708	300.7264269	896
1000-Period					394935.1758	394.9351758	N/A
GS-1000	48.4161586	35.31740578	32.34857842	29.35837578	298979.9095	298.9799095	849
ROIP-10	43.98153542	27.30970603	30.64688325	28.86102228			
200-Period					77767.68438	388.8384219	N/A
GS-200	48.30654808	33.61883439	33.09176045	30.7145995	59780.46454	298.9023227	982
400-Period					160318.6957	400.7967392	N/A
GS-400	48.98942198	33.58660688	32.66001877	30.31229522	120290.5708	300.7264269	896
1000-Period					394935.1758	394.9351758	N/A
GS-1000	48.4161586	35.31740578	32.34857842	29.35837578	298979.9095	298.9799095	849
ROIP-20	48.78629218	40.46256308	25.98808188	40.46256308			
200-Period					56749.15925	283.7457963	N/A
GS-200	46.78629218	38.96256308	25.98808188	39.79589641	59470.49078	297.3524539	5
400-Period					113110.001	282.7750025	N/A
GS-400	47.53104742	39.46256308	26.80157394	37.38889403	118088.2109	295.2205272	26
1000-Period					283759.7892	283.7597892	N/A
GS-1000	47.25382465	40.46256308	25.98808188	36.89034086	294520.7224	294.5207224	12
ROIP-30	48.78629218	40.63579618	25.98808188	40.46256308			
200-Period					56779.28153	283.8964077	N/A
GS-200	46.78629218	38.96912952	25.98808188	39.56256308	59470.15556	297.3507778	6
400-Period					113181.0599	282.9526498	N/A
GS-400	46.78629218	39.63579618	25.98808188	39.79589641	118954.6464	297.3866159	4
1000-Period					283945.0096	283.9450096	N/A
GS-1000	47.03629218	40.63579618	25.98808188	37.27684879	294505.3763	294.5053763	9
ROIP-50	49.67210715	39.74998121	25.98808188	40.46256308			
200-Period					56847.88532	284.2394266	N/A
GS-200	46.67210715	39.08331454	25.98808188	39.56256308	59474.02347	297.3701173	6
400-Period					113301.1888	283.2529721	N/A
GS-400	47.67210715	39.74998121	25.98808188	40.46256308	118368.1518	295.9203795	2
1000-Period					284101.7324	284.1017324	N/A
GS-1000	47.67210715	39.74998121	26.37697077	36.64503061	294404.321	294.404321	13

Table A.17: Normal(32, 8): Experiment Run 7

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
Experiment Run 7							
ROIP-0	0	0	0	0			
200-Period					957414.6936	4787.073468	N/A
GS-200	16.03178657	0	0	0	833226.3313	4166.131657	1701
400-Period					1921829.833	4804.574582	N/A
GS-400	16.02943293	0	0	0	1677252.056	4193.13014	1699
1000-Period					4796887.337	4796.887337	N/A
GS-1000	16.03648554	0	0	0	4213034.934	4213.034934	1705
ROIP-5	46.02970386	35.91432337	40.82530924	20.94394601			
200-Period					60279.28509	301.3964254	N/A
GS-200	46.02970386	35.91432337	40.82530924	28.23466342	58422.08387	292.1104193	22
400-Period					119759.9588	299.399897	N/A
GS-400	46.02970386	35.91432337	40.82530924	28.8672536	116556.6012	291.3915029	30
1000-Period					305298.6956	305.2986956	N/A
GS-1000	46.02970386	35.91432337	40.82530924	28.9984364	292610.5682	292.6105682	32
ROIP-10	46.02970386	38.50965918	38.22997343	38.50965918			
200-Period					58314.06142	291.5703071	N/A
GS-200	44.42970386	37.34299251	37.22997343	38.50965918	58781.7706	293.908853	6
400-Period					116178.7985	290.4469963	N/A
GS-400	46.0681654	35.83743696	36.56330676	33.69978518	116331.8966	290.8297416	56
1000-Period					296275.4146	296.2754146	N/A
GS-1000	46.41154077	36.50965918	36.56330676	33.30013632	291550.8858	291.5508858	58
ROIP-20	46.02970386	35.91432337	40.82530924	35.91432337			
200-Period					57928.69332	289.6434666	N/A
GS-200	46.02970386	35.91432337	37.82530924	35.91432337	58368.02305	291.8401153	3
400-Period					115382.8265	288.4570663	N/A
GS-400	46.02970386	35.41432337	37.15864257	33.57786538	116430.8578	291.0771444	15
1000-Period					294519.5128	294.5195128	N/A
GS-1000	46.43453041	35.91432337	37.15864257	33.10671154	291733.614	291.733614	18
ROIP-30	46.5418678	35.36622674	40.82530924	35.91432337			
200-Period					57882.70786	289.4135393	N/A
GS-200	46.5418678	35.36622674	40.82530924	35.91432337	58851.16992	294.2558496	1
400-Period					115296.6932	288.2417331	N/A
GS-400	46.5418678	35.36622674	37.82530924	35.91432337	116603.9229	291.5098073	3
1000-Period					294348.5651	294.3485651	N/A
GS-1000	46.80972495	35.36622674	37.15864257	33.08738919	291743.4784	291.7434784	17
ROIP-50	47.19332567	31.44200206	42.5982058	30.86387572			
200-Period					58012.59603	290.0629802	N/A
GS-200	47.19332567	33.10866873	40.5982058	30.86387572	58515.58487	292.5779244	4
400-Period					115041.4148	287.6035371	N/A
GS-400	47.19332567	33.50866873	40.0982058	30.86387572	116658.1404	291.645351	6
1000-Period					293456.1707	293.4561707	N/A
GS-1000	47.19332567	31.44200206	42.5982058	30.86387572	295530.0187	295.5300187	1

Table A.18: Normal(32, 8): Experiment Run 8

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					987694.3368	4938.471684	N/A
GS-200	15.95994407	0	0	0	833505.2787	4167.526393	1641
400-Period					1931031.511	4827.578779	N/A
GS-400	16.01762304	0	0	0	1677057.671	4192.644177	1689
1000-Period					4788776.237	4788.776237	N/A
GS-1000	16.04000254	0	0	0	4212806.513	4212.806513	1708
ROIP-5	42.0471358	28.96396792	24.84220079	36.97179489			
200-Period					109618.0882	548.0904408	N/A
GS-200	47.10120598	37.30267065	27.80616409	36.97179489	58981.69454	294.9084727	2001
400-Period					194309.489	485.7737224	N/A
GS-400	49.88755521	35.07096199	27.13841191	36.97179489	118664.1766	296.6604416	1891
1000-Period					456587.6783	456.5876783	N/A
GS-1000	47.09951676	37.00685529	27.77636541	36.97179489	298446.3777	298.4463777	1699
ROIP-10	42.0471358	37.37066853	30.09959733	28.8348408			
200-Period					78411.419	392.057095	N/A
GS-200	47.27366975	37.66563483	37.52958213	31.09330328	57973.81313	289.8690657	1128
400-Period					143020.9588	357.552397	N/A
GS-400	47.14836734	37.37066853	34.95668852	32.68503673	116194.119	290.4852976	560
1000-Period					340036.77	340.03677	N/A
GS-1000	47.21265174	37.37066853	33.54068849	33.01986207	291734.1905	291.7341905	337
ROIP-20	45.40117627	39.85644097	35.09142758	39.7587517			
200-Period					58892.98616	294.4649308	N/A
GS-200	45.40117627	39.85644097	35.09142758	39.7587517	58805.51801	294.02759	1
400-Period					118893.8516	297.2346289	N/A
GS-400	45.40117627	39.85644097	35.09142758	39.7587517	118081.5935	295.2039838	1
1000-Period					295636.5852	295.6365852	N/A
GS-1000	46.38724398	37.85644097	35.09142758	34.04482217	291549.258	291.549258	44
ROIP-30	46.22371533	39.03390191	35.09142758	39.35666984			
200-Period					58733.15366	293.6657683	N/A
GS-200	46.89038199	39.03390191	36.09142758	37.35666984	58703.45702	293.5172851	4
400-Period					118677.3669	296.6934172	N/A
GS-400	46.22371533	39.03390191	35.09142758	39.35666984	117782.0198	294.4550495	1
1000-Period					294906.3321	294.9063321	N/A
GS-1000	46.80322091	37.03390191	35.09142758	34.30425907	291451.1142	291.4511142	26
ROIP-50	48.11276414	38.27452703	35.43232275	33.62085847			
200-Period					58740.96196	293.7048098	N/A
GS-200	48.11276414	38.27452703	37.43232275	33.62085847	58545.96181	292.7298091	2
400-Period					118444.9711	296.1124278	N/A
GS-400	48.11276414	38.27452703	35.43232275	33.62085847	116749.8967	291.8747417	1
1000-Period					292450.914	292.450914	N/A
GS-1000	46.11276414	38.27452703	35.43232275	33.62085847	291671.7587	291.6717587	2

Table A.19: Normal(32, 8): Experiment Run 9

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					951826.9053	4759.134526	N/A
GS-200	16.0458505	0	0	0	833079.7144	4165.398572	1713
400-Period					1897779.332	4744.44833	N/A
GS-400	16.05865613	0	0	0	1679619.7	4199.049251	1724
1000-Period					4731171.253	4731.171253	N/A
GS-1000	16.06791809	0	0	0	4217922.895	4217.922895	1732
ROIP-5	45.47205228	38.88098128	43.42237569	29.53301169			
200-Period					58424.09051	292.1204525	N/A
GS-200	46.13871895	36.88098128	41.92237569	29.53301169	58272.17476	291.3608738	5
400-Period					117000.9113	292.5022781	N/A
GS-400	45.47205228	35.88098128	43.42237569	29.53301169	117292.3299	293.2308247	3
1000-Period					292645.282	292.645282	N/A
GS-1000	45.47205228	35.88098128	40.76443919	29.53301169	293146.4796	293.1464796	10
ROIP-10	45.47205228	38.88098128	43.42237569	29.53301169			
200-Period					58424.09051	292.1204525	N/A
GS-200	46.13871895	36.88098128	41.92237569	29.53301169	58272.17476	291.3608738	5
400-Period					117000.9113	292.5022781	N/A
GS-400	45.47205228	35.88098128	43.42237569	29.53301169	117292.3299	293.2308247	3
1000-Period					292645.282	292.645282	N/A
GS-1000	45.47205228	35.88098128	40.76443919	29.53301169	293146.4796	293.1464796	10
ROIP-20	46.56343123	40.78092577	37.51555765	29.53301169			
200-Period					58013.49265	290.0674632	N/A
GS-200	46.56343123	40.78092577	37.51555765	29.53301169	58310.79495	291.5539747	1
400-Period					116393.3064	290.9832659	N/A
GS-400	46.56343123	40.78092577	37.51555765	29.53301169	116934.8275	292.3370687	1
1000-Period					290058.0268	290.0580268	N/A
GS-1000	44.56343123	39.78092577	37.51555765	29.53301169	292990.1614	292.9901614	3
ROIP-30	46.56343123	40.78092577	37.51555765	29.53301169			
200-Period					58013.49265	290.0674632	N/A
GS-200	46.56343123	40.78092577	37.51555765	29.53301169	58310.79495	291.5539747	1
400-Period					116393.3064	290.9832659	N/A
GS-400	46.56343123	40.78092577	37.51555765	29.53301169	116934.8275	292.3370687	1
1000-Period					290058.0268	290.0580268	N/A
GS-1000	44.56343123	39.78092577	37.51555765	29.53301169	292990.1614	292.9901614	3
ROIP-50	47.4560811	39.88827591	38.48228697	29.53301169			
200-Period					57998.56888	289.9928444	N/A
GS-200	47.4560811	39.88827591	38.48228697	29.53301169	58440.24983	292.2012492	1
400-Period					116600.1991	291.5004977	N/A
GS-400	45.4560811	39.88827591	38.48228697	29.53301169	116731.132	291.82783	2
1000-Period					290992.4677	290.9924677	N/A
GS-1000	45.4560811	37.72160924	38.48228697	29.53301169	292476.6622	292.4766622	5

Table A.20: Normal(32, 8): Experiment Run 10

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					929877.9277	4649.389638	N/A
GS-200	15.47034587	0.618256106	0	0	833440.6753	4167.203376	1750
400-Period					1890599.761	4726.499402	N/A
GS-400	15.47034587	0.591788517	0	0	1679541.426	4198.853566	1727
1000-Period					4777690.084	4777.690084	N/A
GS-1000	15.47034587	0.573167492	0	0	4211442.853	4211.442853	1711
ROIP-5	36.6734155	31.89775063	30.37281126	16.23542591			
200-Period					126308.874	631.5443701	N/A
GS-200	46.99097535	34.74844591	33.56129233	16.23542591	78369.85558	391.8492779	2001
400-Period					271030.9843	677.5774607	N/A
GS-400	47.00012732	35.31687233	32.98371395	16.23542591	157657.4025	394.1435062	2001
1000-Period					716496.2035	716.4962035	N/A
GS-1000	46.94610846	35.85386462	32.50074051	16.23542591	394650.5981	394.6505981	2001
ROIP-10	43.70451034	33.90697294	35.95445504	33.90697294			
200-Period					53779.28835	268.8964417	N/A
GS-200	43.70451034	33.90697294	35.95445504	33.90697294	60192.12667	300.9606334	1
400-Period					113183.9603	282.9599008	N/A
GS-400	45.70451034	35.40697294	35.28778838	33.90697294	117116.1594	292.7903984	5
1000-Period					295406.8549	295.4068549	N/A
GS-1000	45.85835649	37.09922317	35.95445504	32.37396428	292163.409	292.163409	18
ROIP-20	43.70451034	33.90697294	34.66419318	33.90697294			
200-Period					53864.53901	269.322695	N/A
GS-200	43.70451034	35.90697294	34.66419318	33.90697294	59462.84348	297.3142174	2
400-Period					113902.1264	284.7553159	N/A
GS-400	46.20451034	36.46418044	33.66395615	32.11007539	117430.2535	293.5756337	29
1000-Period					297704.7617	297.7047617	N/A
GS-1000	45.70451034	37.63435389	34.7753043	32.75524885	292381.8986	292.3818986	19
ROIP-30	43.72512678	42.75746306	25.79308663	33.90697294			
200-Period					53745.39811	268.7269906	N/A
GS-200	43.72512678	42.75746306	25.79308663	33.90697294	60172.41794	300.8620897	1
400-Period					113101.9193	282.7547982	N/A
GS-400	45.72512678	42.75746306	26.79308663	33.90697294	117483.6443	293.7091107	3
1000-Period					294834.293	294.834293	N/A
GS-1000	45.72512678	42.75746306	28.35975329	33.90697294	293087.8969	293.0878969	6
ROIP-50	43.72512678	43.72512678	32.93930922	25.79308663			
200-Period					53783.23774	268.9161887	N/A
GS-200	43.72512678	43.72512678	32.93930922	25.79308663	59585.84688	297.9292344	1
400-Period					111866.3737	279.6659342	N/A
GS-400	43.72512678	43.72512678	32.93930922	25.79308663	119359.3655	298.3984137	1
1000-Period					290947.0448	290.9470448	N/A
GS-1000	45.72512678	43.72512678	32.93930922	25.79308663	295484.3875	295.4843875	2

A.3 Normal(32, 12)

Table A.21: Normal(32, 12): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					939419.0328	4697.095164	N/A
GS-200	9.623257718	6.440034752	0	0	839842.9893	4199.214947	1728
400-Period					1889615.167	4724.037918	N/A
GS-400	9.623257718	6.440034752	0	0	1685978.621	4214.946553	1728
1000-Period					4802684.185	4802.684185	N/A
GS-1000	9.591733423	6.438876674	0	0	4232977.015	4232.977015	1700
ROIP-5	42.38936997	32.97945779	40.64067816	27.97423911			
200-Period					83891.87606	419.4593803	N/A
GS-200	57.16077595	33.11604417	40.64067816	27.97423911	68978.02312	344.8901156	970
400-Period					170950.5058	427.3762646	N/A
GS-400	55.17504595	34.90008665	40.64067816	27.97423911	137649.3711	344.1234278	877
1000-Period					447201.7566	447.2017566	N/A
GS-1000	55.76584439	34.55181432	40.64067816	27.97423911	345413.1068	345.4131068	990
ROIP-10	42.38936997	32.97945779	40.64067816	27.97423911			
200-Period					83891.87606	419.4593803	N/A
GS-200	57.16077595	33.11604417	40.64067816	27.97423911	68978.02312	344.8901156	970
400-Period					170950.5058	427.3762646	N/A
GS-400	55.17504595	34.90008665	40.64067816	27.97423911	137649.3711	344.1234278	877
1000-Period					447201.7566	447.2017566	N/A
GS-1000	55.76584439	34.55181432	40.64067816	27.97423911	345413.1068	345.4131068	990
ROIP-20	46.51277954	41.96360666	28.74577229	39.28920326			
200-Period					70562.03268	352.8101634	N/A
GS-200	55.1186461	41.96360666	28.79339134	39.28920326	68353.26173	341.7663086	43
400-Period					146508.7115	366.2717786	N/A
GS-400	54.99697384	41.96360666	29.8402828	39.28920326	136337.8026	340.8445065	68
1000-Period					378703.0384	378.7030384	N/A
GS-1000	55.21604581	41.96360666	31.52078957	39.28920326	341885.5432	341.8855432	175
ROIP-30	47.27334438	47.27334438	25.05760452	27.08273547			
200-Period					72779.77388	363.8988694	N/A
GS-200	54.90526074	47.27334438	29.99101592	27.08273547	68852.33991	344.2616996	301
400-Period					151404.7859	378.5119648	N/A
GS-400	54.82526074	47.27334438	30.24309638	27.08273547	137271.9863	343.1799656	328
1000-Period					395655.1849	395.6551849	N/A
GS-1000	55.12768646	47.27334438	28.35120158	29.76464851	343923.5224	343.9235224	566
ROIP-50	51.55025854	47.27334438	27.08273547	25.05760452			
200-Period					68222.1268	341.110634	N/A
GS-200	54.55025854	47.27334438	30.89704292	27.69982888	68608.91441	343.044572	64
400-Period					142208.1209	355.5203023	N/A
GS-400	54.55025854	47.27334438	30.97701245	28.28377461	136754.9971	341.8874927	89
1000-Period					371620.3155	371.6203155	N/A
GS-1000	54.59820571	47.27334438	30.96184052	31.08203736	342458.8094	342.4588094	365

Table A.22: Normal(32, 12): Experiment Run 2

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					965042.0524	4825.210262	N/A
GS-200	16.01406643	0	0	0	841595.9392	4207.979696	1686
400-Period					1917553.899	4793.884747	N/A
GS-400	16.03413743	0	0	0	1689903.683	4224.759207	1703
1000-Period					4835260.342	4835.260342	N/A
GS-1000	16.01762304	0	0	0	4234113.234	4234.113234	1689
ROIP-5	45.53962982	28.06939928	22.1349558	27.13680078			
200-Period					139326.863	696.6343152	N/A
GS-200	60.05085607	29.18972532	22.86013972	27.13680078	88597.40713	442.9870356	2001
400-Period					270017.3574	675.0433935	N/A
GS-400	59.72635097	29.55370761	22.82066253	27.13680078	177788.8105	444.4720263	2001
1000-Period					716199.3973	716.1993973	N/A
GS-1000	56.45458905	33.51117626	22.1349558	27.13680078	442397.6553	442.3976553	2001
ROIP-10	45.53962982	28.06939928	34.94864208	33.78167734			
200-Period					88856.87955	444.2843978	N/A
GS-200	59.82036144	28.85333077	34.94864208	33.78167734	69603.63306	348.0181653	1049
400-Period					175192.042	437.980105	N/A
GS-400	59.7130115	28.85090304	34.94864208	33.78167734	139753.0645	349.3826612	993
1000-Period					468303.8754	468.3038754	N/A
GS-1000	56.18618993	33.26606254	34.94864208	33.78167734	346186.8755	346.1868755	1548
ROIP-20	45.53962982	28.06939928	34.94864208	33.78167734			
200-Period					88856.87955	444.2843978	N/A
GS-200	59.82036144	28.85333077	34.94864208	33.78167734	69603.63306	348.0181653	1049
400-Period					175192.042	437.980105	N/A
GS-400	59.7130115	28.85090304	34.94864208	33.78167734	139753.0645	349.3826612	993
1000-Period					468303.8754	468.3038754	N/A
GS-1000	56.18618993	33.26606254	34.94864208	33.78167734	346186.8755	346.1868755	1548
ROIP-30	48.45302342	31.59040078	48.45302342	31.59040078			
200-Period					68516.79219	342.5839609	N/A
GS-200	55.60668974	31.59040078	48.45302342	27.24299134	68668.33571	343.3416786	177
400-Period					141218.8706	353.0471764	N/A
GS-400	55.81048546	31.59040078	47.44927349	27.8568447	137599.5386	343.9988464	238
1000-Period					368909.5052	368.9095052	N/A
GS-1000	54.67527052	35.15946332	46.05901469	31.59040078	342060.8015	342.0608015	249
ROIP-50	48.45302342	28.90693915	48.45302342	31.59040078			
200-Period					71149.66347	355.7483173	N/A
GS-200	57.74379668	28.90693915	48.45302342	28.52619737	68823.46953	344.1173476	271
400-Period					145682.9176	364.2072939	N/A
GS-400	57.37559936	28.90693915	48.17086957	28.41809489	138263.0671	345.6576676	274
1000-Period					383059.1149	383.0591149	N/A
GS-1000	54.73339207	34.3788461	47.56806816	31.59040078	343235.3713	343.2353713	312

Table A.23: Normal(32, 12): Experiment Run 3

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					943019.8077	4715.099038	N/A
GS-200	16.06676268	0	0	0	839587.0488	4197.935244	1731
400-Period					1924933.448	4812.333619	N/A
GS-400	16.02589728	0	0	0	1689893.206	4224.733014	1696
1000-Period					4781513.039	4781.513039	N/A
GS-1000	16.04351337	0	0	0	4242082.743	4242.082743	1711
ROIP-5	49.32673605	37.14763909	17.89117036	42.67272918			
200-Period					78050.61125	390.2530563	N/A
GS-200	57.2567837	43.48131374	17.89117036	42.67272918	71184.91289	355.9245645	703
400-Period					183238.2106	458.0955265	N/A
GS-400	56.16329693	41.85697365	22.08097406	42.67272918	140516.9835	351.2924587	1467
1000-Period					432179.5107	432.1795107	N/A
GS-1000	55.32417515	38.19621179	26.17222952	42.67272918	351185.3024	351.1853024	1196
ROIP-10	49.32673605	37.14763909	33.82326029	29.54841388			
200-Period					64287.08063	321.4354032	N/A
GS-200	55.36649074	37.31430576	33.82326029	29.54841388	70330.11501	351.6505751	13
400-Period					152298.4668	380.7461671	N/A
GS-400	55.12115264	37.64862113	38.32822515	31.54561347	137107.8554	342.7696385	338
1000-Period					360292.9648	360.2929648	N/A
GS-1000	54.67949263	37.91114187	35.30927456	33.98973288	342739.1293	342.7391293	232
ROIP-20	56.45915098	53.50437872	9.728137311	50.23539239			
200-Period					64154.46611	320.7723305	N/A
GS-200	56.45915098	53.50437872	10.25392003	39.1714424	70809.96111	354.0498055	185
400-Period					143465.1952	358.662988	N/A
GS-400	56.45915098	53.50437872	13.25966403	42.08193052	138847.4208	347.118552	194
1000-Period					345350.6836	345.3506836	N/A
GS-1000	56.45915098	53.50437872	13.02748876	41.63438889	346569.078	346.569078	216
ROIP-30	56.45915098	43.28999246	42.40656002	17.55696968			
200-Period					62247.61767	311.2380883	N/A
GS-200	54.45915098	40.09872262	42.40656002	16.89030301	71176.06004	355.8803002	11
400-Period					139833.1906	349.5829766	N/A
GS-400	56.45915098	43.28999246	42.40656002	19.55696968	137979.0691	344.9476728	2
1000-Period					338121.7955	338.1217955	N/A
GS-1000	56.45915098	43.28999246	42.40656002	17.55696968	345585.9642	345.5859642	1
ROIP-50	56.45915098	42.23529404	43.46125844	16.50227126			
200-Period					61933.32386	309.6666193	N/A
GS-200	54.45915098	39.33529404	43.46125844	16.50227126	71237.22217	356.1861109	7
400-Period					140348.4669	350.8711671	N/A
GS-400	56.45915098	42.23529404	43.46125844	20.16893793	137786.546	344.4663649	4
1000-Period					338472.1286	338.4721286	N/A
GS-1000	53.80684838	41.12539691	43.46125844	23.19288463	343305.7564	343.3057564	105

Table A.24: Normal(32, 12): Experiment Run 4

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					923839.5569	4619.197785	N/A
GS-200	16.11472387	0	0	0	835253.8731	4176.269365	1773
400-Period					1873631.207	4684.078018	N/A
GS-400	16.08745847	0	0	0	1682990.284	4207.47571	1749
1000-Period					4724481.746	4724.481746	N/A
GS-1000	16.07138029	0	0	0	4226121.683	4226.121683	1735
ROIP-5	53.51626796	3.516267959	8.15523408	41.99177474			
200-Period					255138.2241	1275.691121	N/A
GS-200	53.51626796	19.66784063	8.360397617	41.99177474	175055.3401	875.2767006	2001
400-Period					523508.0575	1308.770144	N/A
GS-400	53.51626796	19.87300417	8.15523408	41.99177474	352768.6491	881.9216228	2001
1000-Period					1327499.838	1327.499838	N/A
GS-1000	53.51626796	19.87300417	8.15523408	41.99177474	885261.3077	885.2613077	2001
ROIP-10	58.50224203	47.21607434	29.58770483	30.44810732			
200-Period					69289.57548	346.4478774	N/A
GS-200	59.28000594	40.44192189	29.58770483	30.44810732	68440.08029	342.2004015	25
400-Period					136266.9508	340.667377	N/A
GS-400	59.65290313	38.8481844	29.58770483	30.44810732	137756.1588	344.3903969	66
1000-Period					345893.2313	345.8932313	N/A
GS-1000	55.50224203	45.09385212	29.58770483	31.18382161	342021.9745	342.0219745	11
ROIP-20	59.46945362	35.81372423	40.02284336	20.01296879			
200-Period					68352.27004	341.7613502	N/A
GS-200	62.13612029	34.31372423	40.02284336	20.01296879	69412.35426	347.0617713	5
400-Period					133828.0577	334.5701443	N/A
GS-400	61.46945362	35.81372423	40.02284336	20.01296879	138734.5128	346.836282	2
1000-Period					344728.4859	344.7284859	N/A
GS-1000	59.46945362	35.81372423	40.02284336	23.67963546	344205.8135	344.2058135	4
ROIP-30	59.46945362	31.9063936	43.93017399	16.10563817			
200-Period					68993.29052	344.9664526	N/A
GS-200	63.13612029	31.9063936	44.83017399	16.10563817	69660.04266	348.3002133	6
400-Period					136152.9124	340.382281	N/A
GS-400	62.46945362	32.92544122	45.09684066	16.10563817	139304.9583	348.2623958	8
1000-Period					351541.611	351.541611	N/A
GS-1000	59.46945362	34.97306027	46.90326201	17.26136279	345299.2644	345.2992644	21
ROIP-50	54.58681881	25.27020906	43.93017399	16.10563817			
200-Period					78526.3951	392.6319755	N/A
GS-200	66.054749	26.86579314	43.93017399	16.10563817	71066.17617	355.3308809	386
400-Period					160132.0448	400.3301119	N/A
GS-400	66.02826689	27.32746244	44.83270599	16.10563817	141310.5825	353.2764562	753
1000-Period					414411.3808	414.4113808	N/A
GS-1000	61.38368152	31.67500522	45.44846413	16.10563817	351972.5132	351.9725132	883

Table A.25: Normal(32, 12): Experiment Run 5

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					1001931.648	5009.65824	N/A
GS-200	15.92675338	0	0	0	838186.0144	4190.930072	1614
400-Period					1962730.269	4906.825673	N/A
GS-400	15.9805628	0	0	0	1689109.896	4222.77474	1658
1000-Period					4849849.753	4849.849753	N/A
GS-1000	16.01050348	0	0	0	4234714.919	4234.714919	1683
ROIP-5	45.11457889	33.74230822	16.30513199	34.61514394			
200-Period					146962.5343	734.8126714	N/A
GS-200	55.1243727	38.14489371	18.2494889	34.61514394	79866.13441	399.330672	2001
400-Period					258685.1846	646.7129614	N/A
GS-400	55.18843697	37.92588072	18.40443762	34.61514394	160791.5935	401.9789838	2001
1000-Period					634057.2914	634.0572914	N/A
GS-1000	57.13208423	37.76714665	16.61952442	34.61514394	403287.8779	403.2878779	2001
ROIP-10	51.15948212	27.69740499	38.98516936	34.61514394			
200-Period					84653.60696	423.2680348	N/A
GS-200	59.54471744	34.36535498	38.98516936	34.61514394	67841.76838	339.2088419	1043
400-Period					152903.0579	382.2576446	N/A
GS-400	57.82353876	34.45240578	38.98516936	34.61514394	136323.031	340.8075775	461
1000-Period					379572.3134	379.5723134	N/A
GS-1000	56.28427476	35.00019219	38.98516936	34.61514394	341266.2654	341.2662654	281
ROIP-20	66.72169174	34.91065017	32.03235031	49.56750098			
200-Period					70864.99248	354.3249624	N/A
GS-200	62.67002499	35.10517155	32.03235031	42.19169773	68515.04037	342.5752019	188
400-Period					141796.3387	354.4908467	N/A
GS-400	60.86375523	34.91065017	32.03235031	43.11532247	137412.1168	343.5302921	265
1000-Period					357898.0467	357.8980467	N/A
GS-1000	59.62621243	34.91065017	32.03235031	43.6308392	343588.3329	343.5883329	380
ROIP-30	59.72722661	41.9051153	32.03235031	33.85732725			
200-Period					68831.49247	344.1574624	N/A
GS-200	59.72722661	41.9051153	32.03235031	33.85732725	68165.26112	340.8263056	1
400-Period					136085.0236	340.212559	N/A
GS-400	57.72722661	41.9051153	32.03235031	34.85732725	136602.7676	341.5069191	3
1000-Period					341760.1599	341.7601599	N/A
GS-1000	56.72722661	41.9051153	32.03235031	33.85732725	341183.7312	341.1837312	3
ROIP-50	59.72722661	41.9051153	37.08740112	22.87299105			
200-Period					69992.57166	349.9628583	N/A
GS-200	59.72722661	41.9051153	37.58740112	28.03092756	68472.92247	342.3646123	10
400-Period					137561.1841	343.9029602	N/A
GS-400	59.72722661	41.9051153	37.08740112	24.87299105	137486.8511	343.7171279	2
1000-Period					344601.6051	344.6016051	N/A
GS-1000	59.72722661	41.9051153	37.08740112	22.87299105	344621.8686	344.6218686	1

Table A.26: Normal(32, 12): Experiment Run 6

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					957031.0151	4785.155075	N/A
GS-200	16.03296235	0	0	0	837786.0151	4188.930075	1702
400-Period					1942053.234	4855.133086	N/A
GS-400	16.0045511	0	0	0	1684658.452	4211.646129	1678
1000-Period					4721499.843	4721.499843	N/A
GS-1000	16.07253303	0	0	0	4219805.925	4219.805925	1736
ROIP-5	46.34285607	30.85929397	36.44417852	30.85929397			
200-Period					86546.29795	432.7314897	N/A
GS-200	55.53008049	36.60670165	36.44417852	30.85929397	68517.1665	342.5858325	983
400-Period					183908.2475	459.7706187	N/A
GS-400	58.80577248	33.66057164	36.44417852	30.85929397	137145.6992	342.8642479	1159
1000-Period					417789.9069	417.7899069	N/A
GS-1000	56.96089954	34.41172942	36.44417852	30.85929397	343918.1206	343.9181206	671
ROIP-10	46.34285607	30.85929397	36.44417852	30.85929397			
200-Period					86546.29795	432.7314897	N/A
GS-200	55.53008049	36.60670165	36.44417852	30.85929397	68517.1665	342.5858325	983
400-Period					183908.2475	459.7706187	N/A
GS-400	58.80577248	33.66057164	36.44417852	30.85929397	137145.6992	342.8642479	1159
1000-Period					417789.9069	417.7899069	N/A
GS-1000	56.96089954	34.41172942	36.44417852	30.85929397	343918.1206	343.9181206	671
ROIP-20	46.34285607	30.85929397	29.32241671	30.85929397			
200-Period					98209.54417	491.0477209	N/A
GS-200	55.70259689	37.66991343	29.32241671	30.85929397	70911.09826	354.5554913	1823
400-Period					211541.8561	528.8546403	N/A
GS-400	58.86282077	34.69606548	29.32241671	30.85929397	141814.3959	354.5359896	2001
1000-Period					475584.9901	475.5849901	N/A
GS-1000	55.84977298	37.2390665	29.32241671	30.85929397	355118.9629	355.1189629	1582
ROIP-30	63.97888011	26.65987684	53.0058917	26.65987684			
200-Period					69937.9343	349.6896715	N/A
GS-200	63.97888011	26.94559112	48.5058917	26.00987684	68762.11771	343.8105886	9
400-Period					142235.0382	355.5875955	N/A
GS-400	63.97888011	26.65987684	51.0058917	26.65987684	138294.4222	345.7360555	2
1000-Period					350730.3814	350.7303814	N/A
GS-1000	61.97888011	27.88706808	47.01014423	26.65987684	343483.9768	343.4839768	57
ROIP-50	55.12060299	37.0540745	51.46997116	26.65987684			
200-Period					69365.98104	346.8299052	N/A
GS-200	55.12060299	37.0540745	46.90330449	26.3265435	68237.91715	341.1895858	7
400-Period					141795.7192	354.489298	N/A
GS-400	57.11782702	37.86622172	46.10174596	26.65987684	137020.871	342.5521774	34
1000-Period					348182.3315	348.1823315	N/A
GS-1000	55.12060299	37.1493126	44.09662623	26.65987684	341280.2731	341.2802731	24

Table A.27: Normal(32, 12): Experiment Run 7

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					971623.1048	4858.115524	N/A
GS-200	15.99738477	0	0	0	841970.3625	4209.851813	1672
400-Period					1934645.575	4836.613938	N/A
GS-400	16.01406643	0	0	0	1689606.298	4224.015744	1686
1000-Period					4748220.86	4748.22086	N/A
GS-1000	16.05981622	0	0	0	4236299.081	4236.299081	1725
ROIP-5	44.92933296	39.29884136	33.32509352	29.11008895			
200-Period					87055.94469	435.2797235	N/A
GS-200	59.01760854	39.29884136	33.45154644	29.11008895	68724.66195	343.6233097	686
400-Period					163586.4806	408.9662014	N/A
GS-400	58.83891838	39.29884136	33.32509352	29.11008895	137746.6261	344.3665653	589
1000-Period					385214.9777	385.2149777	N/A
GS-1000	54.69613702	39.5362992	36.07647703	29.11008895	344085.5988	344.0855988	331
ROIP-10	53.73292297	40.94206527	33.60117085	18.3871977			
200-Period					80082.00482	400.4100241	N/A
GS-200	57.96303947	40.94206527	41.17614743	20.58140606	68959.14649	344.7957324	616
400-Period					151620.0813	379.0502033	N/A
GS-400	58.37016809	40.94206527	33.60117085	27.24987206	137972.2538	344.9306345	480
1000-Period					365475.7547	365.4757547	N/A
GS-1000	55.73292297	40.94206527	36.50117085	26.23678251	344220.2303	344.2202303	330
ROIP-20	53.73292297	40.94206527	33.60117085	38.22952			
200-Period					70259.89294	351.2994647	N/A
GS-200	57.39958963	40.94206527	33.60117085	38.22952	68390.23028	341.9511514	4
400-Period					135454.0433	338.6351082	N/A
GS-400	57.53643494	39.94206527	33.60117085	34.33303473	136684.3888	341.7109721	44
1000-Period					335444.8178	335.4448178	N/A
GS-1000	53.73292297	40.94206527	33.60117085	38.22952	341560.7741	341.5607741	1
ROIP-30	58.06967245	41.24640317	28.96008346	30.31707925			
200-Period					71378.75305	356.8937652	N/A
GS-200	58.06967245	41.24640317	36.59199982	30.31707925	68424.51021	342.122551	26
400-Period					135740.5369	339.3513422	N/A
GS-400	58.06967245	41.24640317	31.96008346	30.98374592	137265.0957	343.1627393	4
1000-Period					337327.6145	337.3276145	N/A
GS-1000	58.06967245	41.24640317	30.96008346	30.31707925	343651.124	343.651124	2
ROIP-50	59.32728079	38.36679725	41.24640317	19.65275713			
200-Period					70747.2809	353.7364045	N/A
GS-200	59.32728079	38.44679725	41.24640317	27.20467349	68575.00572	342.8750286	26
400-Period					135441.595	338.6039875	N/A
GS-400	59.32728079	38.36679725	41.24640317	21.65275713	138104.2962	345.2607406	2
1000-Period					338026.8415	338.0268415	N/A
GS-1000	59.32728079	38.36679725	41.24640317	19.65275713	346420.7878	346.4207878	1

Table A.28: Normal(32, 12): Experiment Run 8

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					965172.0288	4825.860144	N/A
GS-200	16.01287948	0	0	0	836464.1976	4182.320988	1685
400-Period					1934064.387	4835.160967	N/A
GS-400	16.01406643	0	0	0	1684063.349	4210.158372	1686
1000-Period					4879878.582	4879.878582	N/A
GS-1000	15.99618788	0	0	0	4225374.142	4225.374142	1671
ROIP-5	49.3028267	34.90004276	30.88639291	34.90004276			
200-Period					71900.82796	359.5041398	N/A
GS-200	53.46949336	41.14754095	30.88639291	34.90004276	67959.89126	339.7994563	103
400-Period					145220.6727	363.0516818	N/A
GS-400	53.05388571	42.0307818	30.88639291	34.90004276	136364.1868	340.910467	130
1000-Period					375668.1421	375.6681421	N/A
GS-1000	55.70308559	40.30510924	30.88639291	34.90004276	340585.1271	340.5851271	206
ROIP-10	49.43702275	44.40199474	21.25024487	44.43446441			
200-Period					68358.27729	341.7913865	N/A
GS-200	53.10368942	44.40199474	26.03407367	41.78396554	68135.09659	340.675483	145
400-Period					138549.7802	346.3744505	N/A
GS-400	53.10368942	44.90199474	26.04971743	42.48815378	136725.4197	341.8135493	132
1000-Period					358766.6329	358.7666329	N/A
GS-1000	55.21277347	45.65771495	23.64411613	42.32502226	342338.6537	342.3386537	180
ROIP-20	57.69866638	53.41906394	28.20938422	38.38074885			
200-Period					69037.74202	345.1887101	N/A
GS-200	53.29984384	45.2065041	28.20938422	38.38074885	67624.85888	338.1242944	308
400-Period					138673.4746	346.6836864	N/A
GS-400	53.22260175	45.50352088	28.20938422	38.38074885	135857.4715	339.6436787	276
1000-Period					354150.9913	354.1509913	N/A
GS-1000	54.19866638	45.54668718	28.20938422	38.38074885	340277.6194	340.2776194	166
ROIP-30	57.69866638	53.41906394	28.20938422	38.38074885			
200-Period					69037.74202	345.1887101	N/A
GS-200	53.29984384	45.2065041	28.20938422	38.38074885	67624.85888	338.1242944	308
400-Period					138673.4746	346.6836864	N/A
GS-400	53.22260175	45.50352088	28.20938422	38.38074885	135857.4715	339.6436787	276
1000-Period					354150.9913	354.1509913	N/A
GS-1000	54.19866638	45.54668718	28.20938422	38.38074885	340277.6194	340.2776194	166
ROIP-50	59.41484351	51.10760916	22.92504909	43.66508398			
200-Period					67717.88316	338.5894158	N/A
GS-200	53.24614912	48.60534177	22.92504909	42.08440542	68071.11396	340.3555698	95
400-Period					136356.611	340.8915274	N/A
GS-400	53.37359946	49.5392184	22.92504909	39.43083808	136286.8228	340.7170569	210
1000-Period					348666.5548	348.6665548	N/A
GS-1000	54.84817684	48.60553634	22.92504909	39.21021231	341228.1909	341.2281909	179

Table A.29: Normal(32, 12): Experiment Run 9

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					975936.8771	4879.684385	N/A
GS-200	15.98778955	0	0	0	836884.5953	4184.422977	1664
400-Period					1946010.673	4865.026683	N/A
GS-400	15.9997764	0	0	0	1686549.747	4216.374367	1674
1000-Period					4852997.966	4852.997966	N/A
GS-1000	16.00931442	0	0	0	4233969.193	4233.969193	1682
ROIP-5	36.52753127	24.42386313	14.36953942	32.25433191			
200-Period					252186.0327	1260.930163	N/A
GS-200	52.88426748	24.42386313	14.36953942	32.25433191	142019.9304	710.0996518	2001
400-Period					492208.8605	1230.522151	N/A
GS-400	52.88426748	24.42386313	14.36953942	32.25433191	286690.8862	716.7272154	2001
1000-Period					1205714.78	1205.71478	N/A
GS-1000	52.88426748	24.42386313	14.36953942	32.25433191	722785.8045	722.7858045	2001
ROIP-10	41.07053245	33.14554731	23.53264773	25.79844404			
200-Period					152999.7947	764.9989736	N/A
GS-200	57.42726866	33.14554731	23.53264773	25.79844404	85042.61545	425.2130773	2001
400-Period					290965.7321	727.4143303	N/A
GS-400	55.26747265	35.30534332	23.53264773	25.79844404	172142.5835	430.3564588	2001
1000-Period					707594.1977	707.5941977	N/A
GS-1000	55.34561035	35.22720562	23.53264773	25.79844404	434744.4786	434.7444786	2001
ROIP-20	55.39249672	44.03320283	33.21374201	21.93037426			
200-Period					72424.82936	362.1241468	N/A
GS-200	55.39249672	46.03320283	33.21374201	31.86751179	67688.26603	338.4413301	220
400-Period					140137.1206	350.3428016	N/A
GS-400	55.39249672	44.43320283	33.21374201	32.71273544	136047.19	340.1179751	151
1000-Period					355372.5796	355.3725796	N/A
GS-1000	55.39249672	44.03320283	33.21374201	33.11273544	341063.8484	341.0638484	151
ROIP-30	55.39249672	49.75161955	27.49532529	27.64879098			
200-Period					69631.29335	348.1564667	N/A
GS-200	55.39249672	49.75161955	28.58952923	34.73218349	68022.81618	340.1140809	34
400-Period					135812.2208	339.5305521	N/A
GS-400	55.39249672	49.75161955	27.49532529	32.21545765	136510.1347	341.2753368	6
1000-Period					346975.6876	346.9756876	N/A
GS-1000	55.39249672	49.75161955	27.49532529	32.21545765	342420.3791	342.4203791	6
ROIP-50	57.67542478	47.46869149	29.89379427	41.06655096			
200-Period					68975.66581	344.878329	N/A
GS-200	55.67542478	46.80202483	29.89379427	36.85411258	67891.80089	339.4590044	18
400-Period					137235.9972	343.089993	N/A
GS-400	54.67542478	46.30202483	30.55392321	34.79235223	136129.5017	340.3237542	145
1000-Period					350108.6053	350.1086053	N/A
GS-1000	54.67542478	45.56869149	30.29686347	36.85106913	341336.4859	341.3364859	66

Table A.30: Normal(32, 12): Experiment Run 10

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					930821.6268	4654.108134	N/A
GS-200	16.0965883	0	0	0	838705.7483	4193.528741	1757
400-Period					1848518.88	4621.2972	N/A
GS-400	16.11923219	0	0	0	1682280.893	4205.702232	1777
1000-Period					4656812.716	4656.812716	N/A
GS-1000	16.10454286	0	0	0	4231552.877	4231.552877	1764
ROIP-5	44.64328586	40.28840632	26.97365851	39.26133682			
200-Period					68274.88528	341.3744264	N/A
GS-200	51.06714986	40.28840632	27.29625259	38.15345281	70986.43226	354.9321613	29
400-Period					137986.3886	344.9659714	N/A
GS-400	51.17238102	40.28840632	28.20244681	38.18584578	140927.689	352.3192226	47
1000-Period					358632.2414	358.6322414	N/A
GS-1000	51.93400327	40.28840632	31.29423731	39.26133682	345114.1764	345.1141764	187
ROIP-10	44.64328586	40.28840632	26.97365851	39.26133682			
200-Period					68274.88528	341.3744264	N/A
GS-200	51.06714986	40.28840632	27.29625259	38.15345281	70986.43226	354.9321613	29
400-Period					137986.3886	344.9659714	N/A
GS-400	51.17238102	40.28840632	28.20244681	38.18584578	140927.689	352.3192226	47
1000-Period					358632.2414	358.6322414	N/A
GS-1000	51.93400327	40.28840632	31.29423731	39.26133682	345114.1764	345.1141764	187
ROIP-20	44.66542339	29.20876297	38.03116433	24.1346731			
200-Period					84293.47532	421.4673766	N/A
GS-200	54.30252548	31.87420015	38.03116433	26.95321817	72624.80945	363.1240473	1079
400-Period					164447.8368	411.1195921	N/A
GS-400	54.02516422	33.06180719	38.03116433	25.57501058	145720.7082	364.3017704	854
1000-Period					432725.8381	432.7258381	N/A
GS-1000	55.35273743	32.88668053	38.03116433	25.02325045	362897.6163	362.8976163	1153
ROIP-30	44.66542339	29.20876297	38.03116433	25.74997923			
200-Period					81725.45146	408.6272573	N/A
GS-200	54.27353997	31.89310448	38.03936108	27.75808729	72114.48337	360.5724169	719
400-Period					160144.989	400.3624726	N/A
GS-400	51.86090271	35.78242105	38.05978406	26.04980003	144306.4911	360.7662279	647
1000-Period					422002.638	422.002638	N/A
GS-1000	55.17484363	33.25389193	38.03116433	25.97345125	359120.8742	359.1208742	909
ROIP-50	52.52070074	32.41310336	26.97154659	22.87842575			
200-Period					83832.33787	419.1616894	N/A
GS-200	54.23220208	39.42607426	30.53017284	26.02669884	72761.48387	363.8074194	1260
400-Period					162206.0404	405.515101	N/A
GS-400	52.69847204	40.88543376	30.53567498	25.57012404	146666.0328	366.6650819	969
1000-Period					426313.9152	426.3139152	N/A
GS-1000	54.75179825	34.69712469	35.55551079	25.30681956	366123.1196	366.1231196	1322

A.4 Beta(2, 5, 32, 8)

Table A.31: Beta(2, 5, 32, 8): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					978545.9519	4892.72976	N/A
GS-200	15.98176908	0	0	0	864052.5582	4320.262791	1659
400-Period					1922888.042	4807.220106	N/A
GS-400	16.02825508	0	0	0	1692658.186	4231.645466	1698
1000-Period					4824044.834	4824.044834	N/A
GS-1000	16.0235367	0	0	0	4247774.36	4247.77436	1694
ROIP-5	53.16118132	26.0057717	33.6574455	31.81862922			
200-Period					67758.44369	338.7922185	N/A
GS-200	53.16118132	35.04339806	33.6574455	31.81862922	62907.81322	314.5390661	52
400-Period					126876.858	317.192145	N/A
GS-400	53.16118132	34.04402696	31.79621579	31.11776983	121233.3743	303.0834358	113
1000-Period					319241.3541	319.2413541	N/A
GS-1000	53.16118132	33.71461113	33.6574455	31.81862922	301343.6935	301.3436935	27
ROIP-10	53.16118132	48.85955207	46.62300207	38.19236395			
200-Period					71091.28309	355.4564154	N/A
GS-200	48.99451465	38.9776852	46.41923375	38.19236395	64562.65954	322.8132977	699
400-Period					143861.5639	359.6539097	N/A
GS-400	48.59451465	38.61831457	46.45796652	38.19236395	128635.9868	321.589967	1002
1000-Period					360233.9052	360.2339052	N/A
GS-1000	49.31899754	37.88688073	46.34478451	38.19236395	320008.2546	320.0082546	1064
ROIP-20	53.16118132	48.85955207	46.62300207	26.0057717			
200-Period					67601.1417	338.0057085	N/A
GS-200	48.99451465	39.76996294	46.62300207	26.0057717	63415.20886	317.0760443	425
400-Period					136620.5449	341.5513623	N/A
GS-400	48.43710724	39.04875276	46.62300207	26.0057717	124169.1335	310.4228337	805
1000-Period					341293.4114	341.2934114	N/A
GS-1000	49.33564802	38.10352682	46.62300207	26.0057717	307552.716	307.552716	824
ROIP-30	53.16118132	48.85955207	46.62300207	26.0057717			
200-Period					67601.1417	338.0057085	N/A
GS-200	48.99451465	39.76996294	46.62300207	26.0057717	63415.20886	317.0760443	425
400-Period					136620.5449	341.5513623	N/A
GS-400	48.43710724	39.04875276	46.62300207	26.0057717	124169.1335	310.4228337	805
1000-Period					341293.4114	341.2934114	N/A
GS-1000	49.33564802	38.10352682	46.62300207	26.0057717	307552.716	307.552716	824
ROIP-50	50.17791593	51.84281746	40.96169221	26.0057717			
200-Period					65596.4933	327.9824665	N/A
GS-200	48.17791593	41.32138956	40.96169221	25.8455225	62960.10834	314.8005417	319
400-Period					131739.6513	329.3491283	N/A
GS-400	48.17791593	40.0236036	40.96169221	26.0057717	122281.0978	305.7027445	563
1000-Period					328538.0272	328.5380272	N/A
GS-1000	48.17791593	40.04505152	40.96169221	26.0057717	303300.3215	303.3003215	557

Table A.32: Beta(2, 5, 32, 8): Experiment Run 2

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					937017.165	4685.085825	N/A
GS-200	16.08173109	0	0	0	821807.6434	4109.038217	1744
400-Period					1882753.408	4706.883519	N/A
GS-400	16.07713735	0	0	0	1651821.407	4129.553517	1740
1000-Period					4730234.919	4730.234919	N/A
GS-1000	16.06907282	0	0	0	4152326.784	4152.326784	1733
ROIP-5	39.12467415	37.07811157	36.82634124	28.62942154			
200-Period					60654.67729	303.2733865	N/A
GS-200	46.5325766	36.16920353	35.88512529	28.62942154	56278.01057	281.3900528	58
400-Period					122958.2037	307.3955093	N/A
GS-400	47.21128999	37.07811157	34.3666697	28.62942154	113210.7036	283.0267589	110
1000-Period					320683.6355	320.6836355	N/A
GS-1000	47.73969063	37.07811157	35.21099399	28.62942154	291421.915	291.421915	94
ROIP-10	47.59753329	35.12281254	35.07483831	31.1007041			
200-Period					56273.64771	281.3682385	N/A
GS-200	47.59753329	35.12281254	35.07483831	31.1007041	56273.64771	281.3682385	1
400-Period					113357.2669	283.3931673	N/A
GS-400	48.59753329	35.12281254	32.40817164	31.1007041	113095.7915	282.7394788	4
1000-Period					291811.6203	291.8116203	N/A
GS-1000	47.59753329	35.12281254	35.07483831	31.1007041	291811.6203	291.8116203	1
ROIP-20	47.59753329	35.12281254	35.07483831	27.72190343			
200-Period					56282.52366	281.4126183	N/A
GS-200	47.59753329	35.12281254	35.07483831	29.72190343	56216.25252	281.0812626	2
400-Period					113324.6197	283.3115494	N/A
GS-400	49.59753329	35.12281254	35.07483831	27.72190343	113175.2807	282.9382018	2
1000-Period					293205.3982	293.2053982	N/A
GS-1000	49.59753329	35.12281254	35.07483831	27.72190343	291189.0057	291.1890057	2
ROIP-30	47.59753329	35.12281254	35.07483831	31.1007041			
200-Period					56273.64771	281.3682385	N/A
GS-200	47.59753329	35.12281254	35.07483831	31.1007041	56273.64771	281.3682385	1
400-Period					113357.2669	283.3931673	N/A
GS-400	48.59753329	35.12281254	32.40817164	31.1007041	113095.7915	282.7394788	4
1000-Period					291811.6203	291.8116203	N/A
GS-1000	47.59753329	35.12281254	35.07483831	31.1007041	291811.6203	291.8116203	1
ROIP-50	47.02395131	35.69639451	35.07483831	27.14832146			
200-Period					56322.45895	281.6122947	N/A
GS-200	47.02395131	35.69639451	35.07483831	29.14832146	56221.02712	281.1051356	2
400-Period					113462.8374	283.6570935	N/A
GS-400	49.02395131	35.69639451	35.07483831	27.14832146	113153.0136	282.882534	2
1000-Period					293986.485	293.986485	N/A
GS-1000	49.02395131	35.69639451	35.07483831	28.14832146	291163.077	291.163077	3

Table A.33: Beta(2, 5, 32, 8): Experiment Run 3

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					942416.5073	4712.082536	N/A
GS-200	16.06791809	0	0	0	827305.9421	4136.52971	1732
400-Period					1925980.807	4814.952018	N/A
GS-400	16.02471734	0	0	0	1695801.767	4239.504418	1695
1000-Period					4817026.275	4817.026275	N/A
GS-1000	16.02707653	0	0	0	4240628.494	4240.628494	1697
ROIP-5	30.97687559	25.01665372	23.39810612	19.81198825			
200-Period					238803.6705	1194.018353	N/A
GS-200	46.40523528	25.94503025	23.39810612	19.81198825	136803.9803	684.0199014	2001
400-Period					508378.6533	1270.946633	N/A
GS-400	46.40176514	25.94850039	23.39810612	19.81198825	298038.7921	745.0969802	2001
1000-Period					1260218.69	1260.21869	N/A
GS-1000	46.38428522	25.9659803	23.39810612	19.81198825	741832.7017	741.8327017	2001
ROIP-10	41.96627019	41.96627019	34.67445343	29.54275236			
200-Period					60683.72645	303.4186322	N/A
GS-200	48.43786196	37.87294319	34.96020384	29.54275236	58461.25801	292.3062901	128
400-Period					125856.9631	314.6424078	N/A
GS-400	48.71268765	39.35052105	34.67445343	30.16062592	120000.5585	300.0013962	83
1000-Period					318044.9118	318.0449118	N/A
GS-1000	48.35358022	40.79404709	34.67445343	30.10020962	303155.2464	303.1552464	33
ROIP-20	42.50060312	41.96627019	39.56453381	34.67445343			
200-Period					60778.93234	303.8946617	N/A
GS-200	46.64134	38.65215801	35.63387716	33.32866714	58795.67418	293.9783709	327
400-Period					125151.9247	312.8798116	N/A
GS-400	47.30212803	38.70552452	36.52581006	34.67445343	120656.3601	301.6409003	145
1000-Period					313355.8179	313.3558179	N/A
GS-1000	46.86603393	40.6477415	35.06949061	34.09919064	303958.7737	303.9587737	122
ROIP-30	41.96627019	41.96627019	40.09886675	34.67445343			
200-Period					61142.04898	305.7102449	N/A
GS-200	46.42607162	38.83695549	35.84056249	34.47828118	58948.51255	294.7425627	232
400-Period					126004.015	315.0100375	N/A
GS-400	46.90244083	38.93432169	36.75596344	34.67445343	120805.9908	302.0149771	161
1000-Period					315204.0975	315.2040975	N/A
GS-1000	46.77349886	40.63877218	35.13045605	34.28787869	304071.138	304.071138	176
ROIP-50	46.44487626	34.26326754	34.67445343	35.02893032			
200-Period					59368.21116	296.8410558	N/A
GS-200	50.53463095	34.26326754	34.67445343	32.12846885	58332.77765	291.6638883	19
400-Period					122599.5435	306.4988587	N/A
GS-400	50.33376515	35.16326754	34.67445343	33.31469455	119719.4176	299.2985441	15
1000-Period					310382.7347	310.3827347	N/A
GS-1000	49.57450589	37.40636545	34.67445343	32.79457178	302717.7132	302.7177132	40

Table A.34: Beta(2, 5, 32, 8): Experiment Run 4

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					954140.491	4770.702455	N/A
GS-200	16.04000254	0	0	0	839229.9128	4196.149564	1708
400-Period					1897405.065	4743.512662	N/A
GS-400	16.05865613	0	0	0	1666738.528	4166.846321	1724
1000-Period					4838986.652	4838.986652	N/A
GS-1000	16.0164382	0	0	0	4262971.468	4262.971468	1688
ROIP-5	38.60862549	33.59368031	28.48022368	25.73172457			
200-Period					98027.97218	490.1398609	N/A
GS-200	51.41147159	37.14757042	28.48022368	25.73172457	64459.3181	322.2965905	2001
400-Period					179259.7317	448.1493292	N/A
GS-400	50.57344636	37.33744089	29.12837844	25.73172457	121315.8135	303.2895337	2001
1000-Period					516858.0055	516.8580055	N/A
GS-1000	53.02310404	35.53593796	28.48022368	25.73172457	328567.9404	328.5679404	2001
ROIP-10	38.60862549	33.59368031	29.73728164	29.7684293			
200-Period					87797.67676	438.9883838	N/A
GS-200	53.03902293	35.35547147	29.73728164	29.7684293	62100.45402	310.5022701	1843
400-Period					158129.7223	395.3243057	N/A
GS-400	50.92225691	36.41038393	29.73728164	29.7684293	118845.0459	297.1126147	1084
1000-Period					451600.8901	451.6008901	N/A
GS-1000	51.35112556	36.82932831	29.73728164	29.7684293	314519.6979	314.5196979	1656
ROIP-20	48.33697165	38.07956588	33.76478074	22.97658294			
200-Period					64108.16092	320.5408046	N/A
GS-200	51.64259742	43.62749593	33.76478074	24.00366922	59810.42168	299.0521084	79
400-Period					120683.3743	301.7084358	N/A
GS-400	50.33697165	39.07956588	35.33144741	25.70063368	117469.692	293.6742299	22
1000-Period					326959.924	326.959924	N/A
GS-1000	51.33697165	41.58269053	35.71746011	24.93494377	306101.0176	306.1010176	103
ROIP-30	51.43335254	44.92353466	27.08476376	34.64636479			
200-Period					60061.75692	300.3087846	N/A
GS-200	52.21766627	43.74786492	27.47597255	30.46862112	59809.02967	299.0451484	34
400-Period					119260.7491	298.1518728	N/A
GS-400	49.43335254	42.75686799	27.08476376	32.95509495	117579.5083	293.9487707	11
1000-Period					305989.6814	305.9896814	N/A
GS-1000	49.43335254	44.92353466	27.08476376	34.64636479	305626.3431	305.6263431	2
ROIP-50	51.43335254	42.17568997	27.08476376	38.85019587			
200-Period					60222.90363	301.1145181	N/A
GS-200	52.25172968	43.57099359	27.08476376	31.91301584	59892.18799	299.4609399	55
400-Period					118825.9461	297.0648654	N/A
GS-400	49.43335254	41.17568997	27.08476376	34.46856937	117663.4743	294.1586858	23
1000-Period					306279.528	306.279528	N/A
GS-1000	51.43335254	42.17568997	27.08476376	38.85019587	306279.528	306.279528	1

Table A.35: Beta(2, 5, 32, 8): Experiment Run 5

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					931912.7023	4659.563512	N/A
GS-200	16.09430975	0	0	0	816613.0673	4083.065336	1755
400-Period					1885969.737	4714.924342	N/A
GS-400	16.07253303	0	0	0	1655103.872	4137.759681	1736
1000-Period					4751450.992	4751.450992	N/A
GS-1000	16.05865613	0	0	0	4173917.483	4173.917483	1724
ROIP-5	40.54113405	31.42488019	30.2066421	36.13704153			
200-Period					68879.59523	344.3979762	N/A
GS-200	52.54986747	31.42488019	30.2066421	36.13704153	58188.07058	290.9403529	228
400-Period					142560.801	356.4020026	N/A
GS-400	51.3760552	34.02200353	30.2066421	36.13704153	116690.7364	291.726841	464
1000-Period					354321.2564	354.3212564	N/A
GS-1000	49.35103028	35.47094531	30.2066421	36.13704153	290458.5764	290.4585764	348
ROIP-10	40.54113405	31.42488019	30.2066421	36.13704153			
200-Period					68879.59523	344.3979762	N/A
GS-200	52.54986747	31.42488019	30.2066421	36.13704153	58188.07058	290.9403529	228
400-Period					142560.801	356.4020026	N/A
GS-400	51.3760552	34.02200353	30.2066421	36.13704153	116690.7364	291.726841	464
1000-Period					354321.2564	354.3212564	N/A
GS-1000	49.35103028	35.47094531	30.2066421	36.13704153	290458.5764	290.4585764	348
ROIP-20	45.93290633	39.79003125	42.6865715	20.8999169			
200-Period					58585.31904	292.9265952	N/A
GS-200	45.93290633	39.79003125	42.6865715	20.8999169	58585.31904	292.9265952	1
400-Period					117709.7671	294.2744178	N/A
GS-400	47.93290633	38.79003125	42.6865715	20.8999169	117049.7246	292.6243114	3
1000-Period					292576.4535	292.5764535	N/A
GS-1000	45.93290633	39.79003125	42.6865715	20.8999169	292576.4535	292.5764535	1
ROIP-30	45.93290633	39.79003125	45.93290633	21.17527992			
200-Period					58827.37464	294.1368732	N/A
GS-200	47.53290633	37.79003125	44.54401744	20.30623231	58382.82795	291.9141398	11
400-Period					118025.3111	295.0632778	N/A
GS-400	48.58319409	37.79003125	43.417823	20.41734342	116920.1874	292.3004684	30
1000-Period					293928.9265	293.9289265	N/A
GS-1000	47.78773287	37.79003125	42.90862783	21.17527992	291582.8939	291.5828939	18
ROIP-50	46.46601415	39.25692343	45.93290633	21.17527992			
200-Period					58716.16637	293.5808318	N/A
GS-200	47.99934748	37.25692343	44.54401744	20.23956564	58332.60008	291.6630004	11
400-Period					117812.4144	294.531036	N/A
GS-400	46.46601415	39.25692343	45.93290633	21.17527992	117812.4144	294.531036	1
1000-Period					293515.6729	293.5156729	N/A
GS-1000	48.30265887	37.25692343	42.89044601	21.17527992	291409.5391	291.4095391	18

Table A.36: Beta(2, 5, 32, 8): Experiment Run 6

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					951095.0232	4755.475116	N/A
GS-200	16.04701804	0	0	0	836134.186	4180.67093	1714
400-Period					1922204.317	4805.510792	N/A
GS-400	16.02825508	0	0	0	1691974.461	4229.936152	1698
1000-Period					4822946.377	4822.946377	N/A
GS-1000	16.0235367	0	0	0	4246675.903	4246.675903	1694
ROIP-5	40.08556875	36.47508258	35.61637103	37.37714297			
200-Period					64241.81974	321.2090987	N/A
GS-200	48.0643948	36.25824049	33.04138023	37.37714297	58066.52274	290.3326137	123
400-Period					124532.0547	311.3301367	N/A
GS-400	47.86665762	36.47508258	32.99295957	35.83541486	114190.3487	285.4758716	221
1000-Period					328388.5174	328.3885174	N/A
GS-1000	48.16500887	36.47508258	35.61637103	34.23474709	299070.381	299.070381	154
ROIP-10	40.08556875	36.47508258	35.61637103	37.37714297			
200-Period					64241.81974	321.2090987	N/A
GS-200	48.0643948	36.25824049	33.04138023	37.37714297	58066.52274	290.3326137	123
400-Period					124532.0547	311.3301367	N/A
GS-400	47.86665762	36.47508258	32.99295957	35.83541486	114190.3487	285.4758716	221
1000-Period					328388.5174	328.3885174	N/A
GS-1000	48.16500887	36.47508258	35.61637103	34.23474709	299070.381	299.070381	154
ROIP-20	50.12335934	27.92868519	34.12497783	37.37714297			
200-Period					59371.11664	296.8555832	N/A
GS-200	53.790026	28.32868519	33.2916445	37.37714297	57721.98099	288.609905	7
400-Period					117162.506	292.9062649	N/A
GS-400	52.12335934	30.81587981	34.12497783	34.55549815	113811.8202	284.5295505	27
1000-Period					312818.0127	312.8180127	N/A
GS-1000	50.12335934	35.29357595	34.12497783	34.53317536	298804.0728	298.8040728	93
ROIP-30	50.12335934	34.68712147	29.96574646	35.43775191			
200-Period					57712.72987	288.5636493	N/A
GS-200	52.12335934	34.68712147	29.96574646	35.43775191	57551.24519	287.756226	2
400-Period					113953.5195	284.8837986	N/A
GS-400	50.12335934	34.68712147	31.96574646	35.43775191	113729.0384	284.3225961	2
1000-Period					302493.7714	302.4937714	N/A
GS-1000	50.12335934	36.68712147	32.54034964	33.9726004	298751.3162	298.7513162	12
ROIP-50	51.08669887	33.12657407	29.96574646	36.78288472			
200-Period					57659.36273	288.2968137	N/A
GS-200	52.08669887	33.12657407	29.96574646	36.78288472	57540.2931	287.7014655	3
400-Period					114048.3112	285.120778	N/A
GS-400	50.08669887	35.12657407	29.96574646	36.11621805	113846.2536	284.6156341	4
1000-Period					303062.1165	303.0621165	N/A
GS-1000	51.08669887	35.63451058	32.45986623	34.58946168	298917.0701	298.9170701	21

Table A.37: Beta(2, 5, 32, 8): Experiment Run 7

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					961562.6249	4807.813125	N/A
GS-200	16.02235537	0	0	0	846778.4711	4233.892355	1693
400-Period					1927700.616	4819.251541	N/A
GS-400	16.02235537	0	0	0	1697555.504	4243.88876	1693
1000-Period					4807306.605	4807.306605	N/A
GS-1000	16.03178657	0	0	0	4230739.433	4230.739433	1701
ROIP-5	42.5679111	33.59135479	21.78558858	31.30811857			
200-Period					93622.92217	468.1146109	N/A
GS-200	51.52186777	37.78248739	24.60531401	31.30811857	58845.0337	294.2251685	1645
400-Period					190986.5085	477.4662713	N/A
GS-400	51.3384103	38.17297046	24.79020992	31.30811857	122795.2645	306.9881612	2001
1000-Period					464841.7383	464.8417383	N/A
GS-1000	50.85107975	37.65128822	25.78718958	31.30811857	310165.3243	310.1653243	1989
ROIP-10	42.5679111	33.59135479	26.05740239	33.23224368			
200-Period					76392.25686	381.9612843	N/A
GS-200	51.8979364	36.32212651	28.06614546	33.23224368	57718.97688	288.5948844	638
400-Period					158602.2444	396.5056111	N/A
GS-400	51.63583698	36.7785593	28.79705901	33.23224368	118901.3534	297.2533836	1013
1000-Period					391109.9998	391.1099998	N/A
GS-1000	50.07557694	38.13551726	28.85919824	33.23224368	301482.1854	301.4821854	944
ROIP-20	42.5679111	35.85422736	32.41914629	39.60357926			
200-Period					60506.01322	302.5300661	N/A
GS-200	49.52100169	35.14522777	32.41914629	38.04273073	57371.73511	286.8586756	57
400-Period					126467.5636	316.168909	N/A
GS-400	50.72428938	35.85422736	32.41914629	36.9569094	117496.9603	293.7424007	125
1000-Period					318430.9128	318.4309128	N/A
GS-1000	50.02974138	35.85422736	32.59588163	37.28649496	298886.0394	298.8860394	82
ROIP-30	45.04725679	38.558883	39.08061234	38.558883			
200-Period					58835.6805	294.1784025	N/A
GS-200	45.04725679	38.33666078	34.33061234	37.67316872	57786.41646	288.9320823	11
400-Period					119926.0509	299.8151271	N/A
GS-400	46.64470183	36.558883	37.50196876	36.35334512	118109.7824	295.2744559	23
1000-Period					303974.4325	303.9744325	N/A
GS-1000	47.47077739	37.15859163	35.45839012	36.42621239	299085.1987	299.0851987	68
ROIP-50	42.98245617	40.62368362	33.47044316	38.558883			
200-Period					58329.79043	291.6489521	N/A
GS-200	47.38723676	36.60734906	33.47044316	37.49221634	57516.50931	287.5825465	65
400-Period					120649.2894	301.6232234	N/A
GS-400	48.50490195	38.41251414	33.47044316	35.87833341	117781.6137	294.4540342	103
1000-Period					305868.8861	305.8688861	N/A
GS-1000	48.1311337	37.90797711	33.47044316	36.49230849	299032.0886	299.0320886	81

Table A.38: Beta(2, 5, 32, 8): Experiment Run 8

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					945530.3228	4727.651614	N/A
GS-200	16.06097564	0	0	0	830469.4933	4152.347467	1726
400-Period					1916653.25	4791.633125	N/A
GS-400	16.03531183	0	0	0	1686322.031	4215.805077	1704
1000-Period					4832832.984	4832.832984	N/A
GS-1000	16.01880717	0	0	0	4256732.603	4256.732603	1690
ROIP-5	38.08112863	33.75514145	32.50942633	29.51790051			
200-Period					81322.04866	406.6102433	N/A
GS-200	46.36951784	38.14756599	35.30395922	29.51790051	60385.10249	301.9255124	1288
400-Period					170224.5477	425.5613693	N/A
GS-400	46.87862325	36.31579818	36.95534538	29.51790051	121752.4378	304.3810945	1518
1000-Period					418240.1382	418.2401382	N/A
GS-1000	49.10893676	36.19534408	34.28129323	29.51790051	303383.9437	303.3839437	1145
ROIP-10	38.37885058	33.45741949	32.50942633	30.54740843			
200-Period					79891.21138	399.4560569	N/A
GS-200	46.31874617	38.2884549	35.0009623	30.54740843	60227.71572	301.1385786	1158
400-Period					166565.2713	416.4131783	N/A
GS-400	47.72208529	36.27394794	35.88469289	30.54740843	121196.9675	302.9924188	1327
1000-Period					409707.05	409.70705	N/A
GS-1000	49.20130283	36.15444582	34.01414705	30.54740843	302097.332	302.097332	1028
ROIP-20	53.97884251	42.06352134	34.90710221	29.85871479			
200-Period					60870.57774	304.3528887	N/A
GS-200	48.59312823	41.40948094	34.90710221	29.85871479	59352.65222	296.7632611	12
400-Period					122709.0732	306.7726829	N/A
GS-400	48.54312823	41.53177531	34.90710221	30.52771246	119426.9028	298.5672569	16
1000-Period					308037.474	308.037474	N/A
GS-1000	48.79312823	39.70488212	34.90710221	31.14773578	298518.6082	298.5186082	47
ROIP-30	53.97884251	42.06352134	34.90710221	29.85871479			
200-Period					60870.57774	304.3528887	N/A
GS-200	48.59312823	41.40948094	34.90710221	29.85871479	59352.65222	296.7632611	12
400-Period					122709.0732	306.7726829	N/A
GS-400	48.54312823	41.53177531	34.90710221	30.52771246	119426.9028	298.5672569	16
1000-Period					308037.474	308.037474	N/A
GS-1000	48.79312823	39.70488212	34.90710221	31.14773578	298518.6082	298.5186082	47
ROIP-50	50.00105449	42.06352134	38.88489024	29.85871479			
200-Period					59978.12122	299.8906061	N/A
GS-200	47.50105449	40.39685467	37.28390984	31.18654705	59238.2465	296.1912325	20
400-Period					120956.2	302.3905001	N/A
GS-400	48.00105449	37.32418687	38.88489024	32.57591213	118756.1292	296.8903231	64
1000-Period					303858.4352	303.8584352	N/A
GS-1000	48.00105449	38.20558483	38.70307205	29.85871479	298641.4125	298.6414125	12

Table A.39: Beta(2, 5, 32, 8): Experiment Run 9

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					952897.1419	4764.485709	N/A
GS-200	16.04234378	0	0	0	837969.7911	4189.848955	1710
400-Period					1918005.441	4795.013602	N/A
GS-400	16.03413743	0	0	0	1687691.091	4219.227727	1703
1000-Period					4774835.059	4774.835059	N/A
GS-1000	16.04701804	0	0	0	4197720.102	4197.720102	1714
ROIP-5	40.88657375	33.52989013	36.68259989	36.70234085			
200-Period					62123.35954	310.6167977	N/A
GS-200	48.2267389	33.52989013	36.39198265	34.08148292	57556.16131	287.7808065	95
400-Period					129914.5569	324.7863923	N/A
GS-400	49.23605487	33.73536195	35.8455275	35.21256812	119742.0507	299.3551269	130
1000-Period					318991.6493	318.9916493	N/A
GS-1000	48.58803566	33.52989013	35.98591709	34.93534436	294650.2709	294.6502709	91
ROIP-10	43.93144905	41.68993824	27.41986456	36.70234085			
200-Period					58585.17866	292.9258933	N/A
GS-200	47.86319508	41.68993824	27.41986456	33.9976289	57933.53833	289.6676916	16
400-Period					122074.6162	305.1865404	N/A
GS-400	48.28383	41.68993824	27.41986456	35.86900752	120118.9579	300.2973948	8
1000-Period					298901.309	298.901309	N/A
GS-1000	47.33144905	41.02327157	27.41986456	35.86900752	294921.4581	294.9214581	7
ROIP-20	41.98926097	41.68993824	27.41986456	36.70234085			
200-Period					59812.02966	299.0601483	N/A
GS-200	46.15592764	41.68993824	27.41986456	35.96900752	58091.65874	290.4582937	7
400-Period					124938.8279	312.3470698	N/A
GS-400	48.26484454	41.68993824	27.41986456	35.98770826	120121.0914	300.3027285	19
1000-Period					305835.7011	305.8357011	N/A
GS-1000	46.98774582	41.68993824	27.41986456	35.66107101	295019.3005	295.0193005	12
ROIP-30	41.68993824	41.68993824	27.41986456	36.70234085			
200-Period					60130.37227	300.6518613	N/A
GS-200	46.18993824	41.68993824	27.41986456	36.01662657	58087.09659	290.435483	8
400-Period					125575.6568	313.9391419	N/A
GS-400	48.33824908	41.68993824	27.41986456	35.96902519	120111.8771	300.2796928	23
1000-Period					307328.0675	307.3280675	N/A
GS-1000	46.94378439	41.68993824	27.41986456	35.5959195	295037.4589	295.0374589	14
ROIP-50	43.0235035	40.27866414	27.41986456	37.07937242			
200-Period					59616.29317	298.0814658	N/A
GS-200	47.9235035	40.27866414	27.41986456	35.61910491	57989.91961	289.949598	14
400-Period					124472.7719	311.1819297	N/A
GS-400	48.20921779	40.27866414	29.38466413	35.88076788	120003.0165	300.0075412	37
1000-Period					304437.2463	304.4372463	N/A
GS-1000	47.81239239	40.27866414	27.60168274	36.0103248	294769.4506	294.7694506	12

Table A.40: Beta(2, 5, 32, 8): Experiment Run 10

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					926771.5761	4633.85788	N/A
GS-200	16.10680979	0	0	0	811382.3907	4056.911954	1766
400-Period					1886557.059	4716.392647	N/A
GS-400	16.07253303	0	0	0	1655691.194	4139.227986	1736
1000-Period					4766788.555	4766.788555	N/A
GS-1000	16.05051658	0	0	0	4189547.777	4189.547777	1717
ROIP-5	30.95337026	28.27094469	28.6239115	19.07883718			
200-Period					173345.3181	866.7265906	N/A
GS-200	47.31010647	28.27094469	28.6239115	19.07883718	92017.88214	460.0894107	2001
400-Period					370614.6232	926.5365579	N/A
GS-400	47.31010647	28.27094469	28.6239115	19.07883718	200507.9546	501.2698864	2001
1000-Period					958927.4476	958.9274476	N/A
GS-1000	47.31010647	28.27094469	28.6239115	19.07883718	524549.1912	524.5491912	2001
ROIP-10	37.88169945	30.02956315	28.78656884	25.71163929			
200-Period					99458.794	497.29397	N/A
GS-200	49.33692806	34.93107075	28.78656884	25.71163929	60846.44053	304.2322026	2001
400-Period					216003.3486	540.0083715	N/A
GS-400	53.53076811	30.7372307	28.78656884	25.71163929	129435.0991	323.5877478	2001
1000-Period					562183.1574	562.1831574	N/A
GS-1000	50.18233127	34.08566754	28.78656884	25.71163929	328819.6846	328.8196846	2001
ROIP-20	39.21300682	29.51827297	30.52424662	35.44649599			
200-Period					78016.12594	390.0806297	N/A
GS-200	49.09364892	33.29779613	30.52424662	35.44649599	58213.78302	291.0689151	520
400-Period					166305.3435	415.7633587	N/A
GS-400	49.29135291	34.27009661	30.52424662	35.44649599	119635.2843	299.0882108	933
1000-Period					426535.142	426.535142	N/A
GS-1000	49.29492685	34.62848299	30.52424662	35.44649599	296889.5521	296.8895521	1118
ROIP-30	45.36589812	31.09794163	35.70528278	38.81917584			
200-Period					60487.15059	302.4357529	N/A
GS-200	49.12488815	32.65719189	35.30261257	33.65828827	58207.56926	291.0378463	130
400-Period					125690.5399	314.2263499	N/A
GS-400	51.63808972	31.84221905	35.70528278	34.53712241	118781.1178	296.9527944	160
1000-Period					309287.1022	309.2871022	N/A
GS-1000	49.26444186	33.78650677	35.32464497	34.7278876	294860.0719	294.8600719	142
ROIP-50	45.36589812	29.51827297	37.28495144	36.06499526			
200-Period					61173.07376	305.8653688	N/A
GS-200	49.10842445	32.68855904	37.0984839	31.8909896	58188.39493	290.9419746	158
400-Period					127534.0242	318.8350605	N/A
GS-400	51.42925225	31.56409382	36.97897565	34.06598116	118887.7117	297.2192792	103
1000-Period					315022.4811	315.0224811	N/A
GS-1000	49.24813187	33.77839716	35.4595406	34.64833909	294880.0004	294.8800004	167

A.5 Beta(2, 5, 32, 12)

Table A.41: Beta(2, 5, 32, 12): Experiment Run 1

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					1000520.132	5002.600662	N/A
GS-200	15.93046856	0	0	0	886394.2557	4431.971279	1617
400-Period					1943675.546	4859.188866	N/A
GS-400	16.00335849	0	0	0	1713803.305	4284.508263	1677
1000-Period					4817238.015	4817.238015	N/A
GS-1000	16.02707653	0	0	0	4240840.235	4240.840235	1697
ROIP-5	54.66000546	38.57053003	53.89713355	21.90327234			
200-Period					83940.65744	419.7032872	N/A
GS-200	64.22139326	38.64624431	53.89713355	21.90327234	81354.43054	406.7721527	70
400-Period					159947.3417	399.8683541	N/A
GS-400	60.09571975	38.57053003	53.89713355	21.90327234	157977.9066	394.9447664	9
1000-Period					366888.0342	366.8880342	N/A
GS-1000	57.06148711	38.23441891	49.3440089	21.90327234	364708.6455	364.7086455	22
ROIP-10	54.66000546	38.57053003	53.89713355	21.90327234			
200-Period					83940.65744	419.7032872	N/A
GS-200	64.22139326	38.64624431	53.89713355	21.90327234	81354.43054	406.7721527	70
400-Period					159947.3417	399.8683541	N/A
GS-400	60.09571975	38.57053003	53.89713355	21.90327234	157977.9066	394.9447664	9
1000-Period					366888.0342	366.8880342	N/A
GS-1000	57.06148711	38.23441891	49.3440089	21.90327234	364708.6455	364.7086455	22
ROIP-20	63.42250261	63.42250261	60.37271105	34.75292543			
200-Period					92598.45034	462.9922517	N/A
GS-200	63.42250261	49.47254576	59.56455835	34.75292543	85781.59944	428.9079972	900
400-Period					186979.9465	467.4498663	N/A
GS-400	61.42250261	50.3789327	60.37271105	34.75292543	171587.792	428.96948	1038
1000-Period					468136.1407	468.1361407	N/A
GS-1000	59.2441808	52.16311118	60.37271105	34.75292543	423604.6127	423.6046127	1264
ROIP-30	63.42250261	63.42250261	58.25339143	34.75292543			
200-Period					91789.52567	458.9476284	N/A
GS-200	63.42250261	49.01500002	58.04084961	34.75292543	85035.98249	425.1799125	840
400-Period					185288.499	463.2212475	N/A
GS-400	61.42250261	50.45156089	58.25339143	34.75292543	169977.442	424.943605	1001
1000-Period					463835.407	463.835407	N/A
GS-1000	59.25583594	52.1577977	58.25339143	34.75292543	419352.6898	419.3526898	1260
ROIP-50	63.42250261	45.84638282	47.15993403	45.84638282			
200-Period					82930.58639	414.6529319	N/A
GS-200	62.42250261	40.91944865	44.29697198	44.2221142	81175.84046	405.8792023	103
400-Period					164943.7403	412.3593507	N/A
GS-400	60.75583594	40.2536602	45.27470709	43.8137452	158936.4096	397.3410241	248
1000-Period					406022.2024	406.0222024	N/A
GS-1000	58.79701241	36.70891034	46.7346003	45.84638282	377118.0831	377.1180831	677

Table A.42: Beta(2, 5, 32, 12): Experiment Run 2

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					922097.3881	4610.486941	N/A
GS-200	16.11923219	0	0	0	806619.2087	4033.096044	1777
400-Period					1819588.867	4548.972167	N/A
GS-400	16.15604304	0	0	0	1587523.465	3968.808662	1810
1000-Period					4734864.746	4734.864746	N/A
GS-1000	16.06676268	0	0	0	4157039.693	4157.039693	1731
ROIP-5	47.9971373	39.18197789	24.26561722	43.93983018			
200-Period					69853.03312	349.2651656	N/A
GS-200	56.46658623	39.18197789	24.38597018	41.48478154	65693.20613	328.4660306	141
400-Period					133546.0106	333.8650265	N/A
GS-400	52.53359863	39.18197789	27.33705038	39.0918258	127671.3932	319.178483	285
1000-Period					392847.9772	392.8479772	N/A
GS-1000	58.07186859	40.17958333	25.21078584	43.93983018	356845.7113	356.8457113	229
ROIP-10	47.9971373	39.18197789	24.28477597	43.92067143			
200-Period					69839.23882	349.1961941	N/A
GS-200	56.44892004	39.18197789	24.4203357	41.44897897	65690.61882	328.4530941	142
400-Period					133524.383	333.8109575	N/A
GS-400	52.51872011	39.18197789	27.35468088	39.09172217	127671.179	319.1779476	280
1000-Period					392770.8275	392.7708275	N/A
GS-1000	58.07186859	40.16618163	25.21691422	43.92067143	356841.5631	356.8415631	226
ROIP-20	47.9971373	31.59858268	36.84409185	31.36135555			
200-Period					70647.2961	353.2364805	N/A
GS-200	55.54905365	32.10115672	36.84409185	31.36135555	64819.74466	324.0987233	32
400-Period					134592.1784	336.480446	N/A
GS-400	52.16380396	34.62739533	36.84409185	31.36135555	126536.6191	316.3415477	21
1000-Period					400983.1248	400.9831248	N/A
GS-1000	59.28308113	32.63030104	36.84409185	31.36135555	354309.1058	354.3091058	266
ROIP-30	52.54047687	35.09342031	25.25326407	31.36135555			
200-Period					70914.37395	354.5718697	N/A
GS-200	54.59568086	35.09342031	34.82005146	31.36135555	64810.27197	324.0513599	188
400-Period					134740.2474	336.8506186	N/A
GS-400	52.54047687	37.09342031	34.00752957	31.36135555	126464.9212	316.162303	122
1000-Period					406658.67	406.65867	N/A
GS-1000	58.23990751	42.72779292	25.40682796	31.36135555	355644.1316	355.6441316	477
ROIP-50	52.54047687	37.43923433	29.72392372	27.39514561			
200-Period					67611.97947	338.0598973	N/A
GS-200	54.54047687	37.43923433	34.39425292	29.55819089	64982.66882	324.9133441	47
400-Period					129744.8718	324.3621795	N/A
GS-400	52.54047687	37.43923433	34.03344753	30.37633921	126547.0825	316.3677062	22
1000-Period					384288.66	384.28866	N/A
GS-1000	57.97619116	41.15438084	31.27732612	28.54428694	354670.134	354.670134	211

Table A.43: Beta(2, 5, 32, 12): Experiment Run 3

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					924516.539	4622.582695	N/A
GS-200	16.1124659	0	0	0	809086.8333	4045.434166	1771
400-Period					1883830.196	4709.575489	N/A
GS-400	16.07598726	0	0	0	1652914.715	4132.286787	1739
1000-Period					4772555.383	4772.555383	N/A
GS-1000	16.0481849	0	0	0	4195398.461	4195.398461	1715
ROIP-5	56.3374965	19.89057778	24.41309427	36.53455652			
200-Period					92613.45873	463.0672937	N/A
GS-200	56.3374965	34.09653435	26.38297745	36.53455652	71024.59005	355.1229502	1828
400-Period					198996.5224	497.4913061	N/A
GS-400	70.66289151	19.89057778	26.35027636	36.53455652	149978.2072	374.9455179	1909
1000-Period					516082.4591	516.0824591	N/A
GS-1000	64.51509295	28.06971754	24.41309427	36.53455652	371038.2458	371.0382458	2001
ROIP-10	56.3374965	19.89057778	32.00933268	29.37703587			
200-Period					87523.95247	437.6197623	N/A
GS-200	56.3374965	33.74852524	33.84713814	29.37703587	69268.05231	346.3402615	1438
400-Period					184852.8379	462.1320948	N/A
GS-400	70.14958658	19.89057778	33.93157749	29.37703587	146033.8415	365.0846038	1466
1000-Period					477508.2251	477.5082251	N/A
GS-1000	60.00416317	32.22249206	32.00933268	29.37703587	359671.0956	359.6710956	1673
ROIP-20	62.15746542	51.50267365	34.29933557	38.94119686			
200-Period					75649.79728	378.2489864	N/A
GS-200	51.69900178	47.17871806	34.29933557	38.94119686	68976.30954	344.8815477	911
400-Period					152186.3606	380.4659014	N/A
GS-400	56.0433557	43.88849423	34.29933557	38.94119686	142217.0936	355.542734	538
1000-Period					378048.7976	378.0487976	N/A
GS-1000	56.49952891	43.37365505	34.29933557	38.94119686	350764.1799	350.7641799	554
ROIP-30	62.15746542	43.05652912	34.29933557	43.68808424			
200-Period					72385.21855	361.9260927	N/A
GS-200	52.22818598	39.1104999	34.29933557	43.68808424	67736.97201	338.6848601	579
400-Period					146819.875	367.0496875	N/A
GS-400	56.97175113	38.68739807	34.29933557	40.84407552	141228.7686	353.0719214	277
1000-Period					363832.7994	363.8327994	N/A
GS-1000	56.72175113	39.35007166	34.29933557	40.505075	348500.24	348.50024	267
ROIP-50	62.15746542	43.05652912	33.2435304	44.74388941			
200-Period					72187.33823	360.9366912	N/A
GS-200	51.82866374	40.16712607	33.2435304	44.74388941	67859.25501	339.296275	417
400-Period					146524.9311	366.3123278	N/A
GS-400	57.25746542	38.61493094	33.2435304	41.77537257	141405.776	353.51444	265
1000-Period					363297.1911	363.2971911	N/A
GS-1000	56.72175113	40.25570636	33.2435304	40.47591414	348679.8012	348.6798012	292

Table A.44: Beta(2, 5, 32, 12): Experiment Run 4

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					945029.8716	4725.149358	N/A
GS-200	16.06213439	0	0	0	829960.7408	4149.803704	1727
400-Period					1923564.089	4808.910222	N/A
GS-400	16.02707653	0	0	0	1693351.162	4233.377904	1697
1000-Period					4817479.251	4817.479251	N/A
GS-1000	16.02589728	0	0	0	4241123.881	4241.123881	1696
ROIP-5	38.20827409	31.5756985	11.92263723	34.61827809			
200-Period					181617.5152	908.0875759	N/A
GS-200	47.36645393	36.55777224	14.13911987	34.61827809	109584.6361	547.9231804	2001
400-Period					377383.9911	943.4599776	N/A
GS-400	54.5650103	31.5756985	11.92263723	34.61827809	228874.1438	572.1853596	2001
1000-Period					941912.2922	941.9122922	N/A
GS-1000	54.5650103	31.5756985	11.92263723	34.61827809	567714.2348	567.7142348	2001
ROIP-10	40.79464165	31.95281103	30.2554662	24.52734601			
200-Period					117911.8167	589.5590835	N/A
GS-200	53.21832279	32.74887828	33.39245401	24.52734601	76400.49246	382.0024623	2001
400-Period					246608.1204	616.5203009	N/A
GS-400	53.7607993	35.34338958	30.2554662	24.52734601	159075.0644	397.687661	2001
1000-Period					610464.3758	610.4643758	N/A
GS-1000	56.74725746	32.35693142	30.2554662	24.52734601	404705.6631	404.7056631	2001
ROIP-20	40.90726282	31.84018985	30.2554662	29.71243105			
200-Period					106305.8943	531.5294713	N/A
GS-200	53.18164761	33.87271424	32.30529324	29.71243105	71319.93158	356.5996579	2001
400-Period					223038.8705	557.5971763	N/A
GS-400	53.82220721	35.28198167	30.2554662	29.71243105	147894.6556	369.736639	2001
1000-Period					555504.8498	555.5048498	N/A
GS-1000	56.63264233	32.47154656	30.2554662	29.71243105	380255.1191	380.2551191	2001
ROIP-30	41.75381811	38.52129214	30.44196331	29.71243105			
200-Period					86371.41994	431.8570997	N/A
GS-200	53.21972667	39.34063952	33.41398402	29.71243105	67572.29166	337.8614583	1155
400-Period					182169.218	455.423045	N/A
GS-400	54.25566703	38.83065398	33.37597551	29.71243105	136944.2694	342.3606734	1474
1000-Period					460370.4584	460.3704584	N/A
GS-1000	55.09425017	38.52129214	32.66945891	29.71243105	355987.321	355.987321	1349
ROIP-50	51.22536459	35.88128579	30.99687681	38.08778927			
200-Period					68806.3617	344.0318085	N/A
GS-200	53.22536459	40.06187988	33.37054274	37.28230846	66323.43405	331.6171703	61
400-Period					142620.8412	356.552103	N/A
GS-400	54.72536459	39.06326287	34.68862688	37.61084723	133778.1056	334.4452641	128
1000-Period					367732.265	367.732265	N/A
GS-1000	55.72536459	38.06673831	34.16642537	38.08778927	346655.2674	346.6552674	78

Table A.45: Beta(2, 5, 32, 12): Experiment Run 5

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					911417.1485	4557.085743	N/A
GS-200	16.14607072	0	0	0	795746.6979	3978.733489	1801
400-Period					1895511.646	4738.779114	N/A
GS-400	16.06097564	0	0	0	1664811.792	4162.029479	1726
1000-Period					4732134.724	4732.134724	N/A
GS-1000	16.06791809	0	0	0	4154268.118	4154.268118	1732
ROIP-5	41.11580565	31.22904109	36.26582605	31.22904109			
200-Period					88835.05684	444.1752842	N/A
GS-200	55.65692006	32.50644233	36.26582605	31.22904109	70107.99429	350.5399714	1529
400-Period					196039.4782	490.0986955	N/A
GS-400	57.41369916	31.22904109	36.26582605	31.22904109	140425.1836	351.062959	1943
1000-Period					488576.5271	488.5765271	N/A
GS-1000	57.34136424	31.22904109	36.26582605	31.22904109	356350.7583	356.3507583	1874
ROIP-10	54.36105891	38.57666371	31.72262172	34.33887446			
200-Period					68918.10273	344.5905136	N/A
GS-200	56.36105891	40.24333038	31.72262172	34.33887446	68470.47919	342.3523959	4
400-Period					138644.2959	346.6107399	N/A
GS-400	58.93566209	40.14848887	31.92416343	31.13799043	135888.1173	339.7202932	67
1000-Period					351824.2304	351.8242304	N/A
GS-1000	57.36105891	38.57666371	35.29522706	33.12856629	345973.7829	345.9737829	28
ROIP-20	54.36105891	38.18724278	38.06471397	27.00615513			
200-Period					69098.86909	345.4943454	N/A
GS-200	54.36105891	38.18724278	38.06471397	31.90615513	68472.36806	342.3618403	7
400-Period					137661.1843	344.1529609	N/A
GS-400	58.83573424	38.18724278	36.03626464	27.00615513	136038.4943	340.0962357	15
1000-Period					350545.565	350.545565	N/A
GS-1000	56.36105891	38.18724278	38.06471397	31.21257649	345938.3194	345.9383194	13
ROIP-30	54.36105891	38.18724278	38.06471397	27.00615513			
200-Period					69098.86909	345.4943454	N/A
GS-200	54.36105891	38.18724278	38.06471397	31.90615513	68472.36806	342.3618403	7
400-Period					137661.1843	344.1529609	N/A
GS-400	58.83573424	38.18724278	36.03626464	27.00615513	136038.4943	340.0962357	15
1000-Period					350545.565	350.545565	N/A
GS-1000	56.36105891	38.18724278	38.06471397	31.21257649	345938.3194	345.9383194	13
ROIP-50	54.36105891	38.57666371	42.19898329	27.44366119			
200-Period					68792.68794	343.9634397	N/A
GS-200	54.36105891	38.57666371	42.19898329	27.44366119	68792.68794	343.9634397	1
400-Period					136933.0055	342.3325137	N/A
GS-400	58.60918523	38.57666371	36.23825125	27.44366119	136033.5001	340.0837502	93
1000-Period					346941.4558	346.9414558	N/A
GS-1000	54.36105891	38.57666371	42.19898329	27.44366119	346941.4558	346.9414558	1

Table A.46: Beta(2, 5, 32, 12): Experiment Run 6

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					953207.9914	4766.039957	N/A
GS-200	16.04234378	0	0	0	838280.6406	4191.403203	1710
400-Period					1893923.826	4734.809565	N/A
GS-400	16.06329247	0	0	0	1663190.693	4157.976732	1728
1000-Period					4894732.844	4894.732844	N/A
GS-1000	15.99019268	0	0	0	4319661.555	4319.661555	1666
ROIP-5	43.85511553	34.14209002	33.64407661	23.63803055			
200-Period					94010.97939	470.0548969	N/A
GS-200	55.15144621	34.14209002	33.64407661	27.52486129	73549.21718	367.7460859	1113
400-Period					185093.8943	462.7347359	N/A
GS-400	55.29459636	34.14209002	33.64407661	27.63784523	144554.1393	361.3853482	1265
1000-Period					532389.5159	532.3895159	N/A
GS-1000	59.73420281	34.61973894	33.64407661	23.63803055	377549.8917	377.5498917	2001
ROIP-10	60.41889548	42.62631105	38.91126791	36.40168345			
200-Period					73849.61901	369.2480951	N/A
GS-200	55.05862797	35.21886676	38.91126791	36.40168345	71141.54173	355.7077086	333
400-Period					145216.9018	363.0422545	N/A
GS-400	55.21247412	35.1014404	38.91126791	36.40168345	138361.9054	345.9047635	327
1000-Period					356734.8803	356.7348803	N/A
GS-1000	58.41889548	40.45964439	35.02546032	35.34033396	350049.5565	350.0495565	54
ROIP-20	62.95321213	55.69817094	52.94325195	50.05343647			
200-Period					88757.31839	443.7865919	N/A
GS-200	55.66249472	47.80213465	52.94325195	50.05343647	81291.03058	406.4551529	1115
400-Period					178819.802	447.0495049	N/A
GS-400	55.75773281	47.44017029	52.94325195	50.05343647	161809.0541	404.5226353	1274
1000-Period					443994.3413	443.9943413	N/A
GS-1000	57.76749784	45.32636651	52.94325195	50.05343647	398635.9007	398.6359007	1342
ROIP-30	62.95321213	55.69817094	52.94325195	23.96692824			
200-Period					82399.85855	411.9992927	N/A
GS-200	55.75773281	47.72311286	52.94325195	23.96692824	75272.42672	376.3621336	1106
400-Period					165173.8554	412.9346386	N/A
GS-400	55.96299597	47.27937104	52.94325195	23.96692824	148518.2206	371.2955514	1246
1000-Period					406907.9822	406.9079822	N/A
GS-1000	58.05321213	45.39533136	52.94325195	23.96692824	365439.7	365.4397	1124
ROIP-50	62.60943204	52.94325195	44.68445483	23.96692824			
200-Period					78089.21466	390.4460733	N/A
GS-200	55.29340673	46.18034748	44.68445483	23.96692824	73101.58873	365.5079437	641
400-Period					155141.3956	387.8534889	N/A
GS-400	55.26461478	45.52995967	44.68445483	23.96692824	142358.9615	355.8974038	900
1000-Period					380147.0797	380.1470797	N/A
GS-1000	58.04276537	43.69782853	44.68445483	23.96692824	353459.0383	353.4590383	561

Table A.47: Beta(2, 5, 32, 12): Experiment Run 7

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					925711.571	4628.557855	N/A
GS-200	16.11020537	0	0	0	810298.0597	4051.490298	1769
400-Period					1929880.185	4824.700461	N/A
GS-400	16.0199906	0	0	0	1699769.039	4249.422599	1691
1000-Period					4870736.912	4870.736912	N/A
GS-1000	16.00097114	0	0	0	4295277.985	4295.277985	1675
ROIP-5	64.58316738	50.8279145	18.85159742	46.55948446			
200-Period					72599.97033	362.9998517	N/A
GS-200	58.96563491	50.8279145	18.4293752	40.64396094	69679.24663	348.3962332	222
400-Period					151658.3421	379.1458553	N/A
GS-400	62.58316738	50.8279145	18.85159742	37.39045662	149225.3641	373.0634103	150
1000-Period					374803.1491	374.8031491	N/A
GS-1000	60.63078643	50.8279145	18.85159742	39.52478038	368030.5816	368.0305816	137
ROIP-10	64.58316738	50.8279145	18.85159742	46.55948446			
200-Period					72599.97033	362.9998517	N/A
GS-200	58.96563491	50.8279145	18.4293752	40.64396094	69679.24663	348.3962332	222
400-Period					151658.3421	379.1458553	N/A
GS-400	62.58316738	50.8279145	18.85159742	37.39045662	149225.3641	373.0634103	150
1000-Period					374803.1491	374.8031491	N/A
GS-1000	60.63078643	50.8279145	18.85159742	39.52478038	368030.5816	368.0305816	137
ROIP-20	63.11058467	32.86643573	33.02795118	32.38313071			
200-Period					69886.79536	349.4339768	N/A
GS-200	63.11058467	36.36643573	33.02795118	28.68765139	69425.41034	347.1270517	21
400-Period					152292.5997	380.7314992	N/A
GS-400	63.11058467	40.33501876	33.02795118	32.38313071	149274.2702	373.1856754	24
1000-Period					371341.8505	371.3418505	N/A
GS-1000	63.11058467	37.4331024	33.02795118	32.38313071	366872.6935	366.8726935	6
ROIP-30	63.11058467	24.4982907	41.39609621	24.01498568			
200-Period					71874.31952	359.3715976	N/A
GS-200	63.11058467	34.4761048	40.97935329	24.01498568	69578.29063	347.8914532	102
400-Period					161171.6369	402.9290923	N/A
GS-400	63.11058467	36.33260203	41.39609621	24.04355754	149756.1599	374.3903997	212
1000-Period					390887.9915	390.8879915	N/A
GS-1000	63.11058467	35.9506592	41.39609621	24.30867908	367869.7867	367.8697867	200
ROIP-50	61.61726338	42.35282318	25.03488503	40.37619685			
200-Period					69982.07015	349.9103507	N/A
GS-200	59.61726338	44.32771866	25.03488503	34.92630656	69541.53042	347.7076521	63
400-Period					150252.0703	375.6301759	N/A
GS-400	62.61726338	45.30962113	25.03488503	36.34302055	149384.9019	373.4622547	31
1000-Period					368310.7288	368.3107288	N/A
GS-1000	61.61726338	42.35282318	27.79781356	35.43028595	366731.5644	366.7315644	27

Table A.48: Beta(2, 5, 32, 12): Experiment Run 8

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					987426.9663	4937.134831	N/A
GS-200	15.96116283	0	0	0	873081.1957	4365.405979	1642
400-Period					1947951.468	4869.87867	N/A
GS-400	15.99858094	0	0	0	1718147.851	4295.369628	1673
1000-Period					4898355.909	4898.355909	N/A
GS-1000	15.98778955	0	0	0	4323371.046	4323.371046	1664
ROIP-5	62.33121328	12.33121328	12.33121328	21.70187916			
200-Period					233179.8474	1165.899237	N/A
GS-200	62.33121328	28.68794949	12.33121328	21.70187916	143665.8084	718.3290418	2001
400-Period					432416.9949	1081.042487	N/A
GS-400	62.33121328	28.68794949	12.33121328	21.70187916	269251.7014	673.1292536	2001
1000-Period					1106290.218	1106.290218	N/A
GS-1000	62.33121328	28.68794949	12.33121328	21.70187916	703733.4477	703.7334477	2001
ROIP-10	61.29918535	35.68593835	26.97505599	45.22049832			
200-Period					73081.81036	365.4090518	N/A
GS-200	61.29918535	38.68593835	32.78622069	43.80129929	71206.1431	356.0307155	94
400-Period					142493.7346	356.2343364	N/A
GS-400	61.29918535	37.68593835	29.50237589	41.57022177	140513.1958	351.2829896	34
1000-Period					377107.2865	377.1072865	N/A
GS-1000	61.29918535	39.35260501	31.4980502	42.86387181	368275.3308	368.2753308	110
ROIP-20	61.29918535	35.68593835	26.97505599	45.22049832			
200-Period					73081.81036	365.4090518	N/A
GS-200	61.29918535	38.68593835	32.78622069	43.80129929	71206.1431	356.0307155	94
400-Period					142493.7346	356.2343364	N/A
GS-400	61.29918535	37.68593835	29.50237589	41.57022177	140513.1958	351.2829896	34
1000-Period					377107.2865	377.1072865	N/A
GS-1000	61.29918535	39.35260501	31.4980502	42.86387181	368275.3308	368.2753308	110
ROIP-30	61.29918535	40.63818146	34.9712293	33.20407612			
200-Period					70838.41541	354.1920771	N/A
GS-200	61.29918535	40.63818146	34.9712293	33.20407612	70838.41541	354.1920771	1
400-Period					139547.7073	348.8692682	N/A
GS-400	59.29918535	41.63818146	34.9712293	33.20407612	139362.4955	348.4062386	3
1000-Period					366479.542	366.479542	N/A
GS-1000	61.29918535	40.63818146	34.9712293	33.20407612	366479.542	366.479542	1
ROIP-50	61.29918535	42.81929036	42.81929036	32.98473234			
200-Period					71814.38032	359.0719016	N/A
GS-200	59.29918535	41.41929036	41.6526237	32.98473234	71100.87647	355.5043824	6
400-Period					143717.3838	359.2934595	N/A
GS-400	57.34680439	40.98211088	38.80613954	32.98473234	139479.0199	348.6975498	76
1000-Period					371653.8417	371.6538417	N/A
GS-1000	57.63251868	40.62802052	40.44080689	32.98473234	365801.7457	365.8017457	35

Table A.49: Beta(2, 5, 32, 12): Experiment Run 9

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					967789.0188	4838.945094	N/A
GS-200	15.06752675	0.928661137	0	0	853225.7606	4266.128803	1671
400-Period					1884229.495	4710.573738	N/A
GS-400	15.06752675	1.002700141	0	0	1653432.853	4133.582134	1734
1000-Period					4710441.166	4710.441166	N/A
GS-1000	15.06752675	1.008460514	0	0	4132320.664	4132.320664	1739
ROIP-5	55.55154955	33.51354007	43.03802657	24.13683875			
200-Period					70771.98228	353.8599114	N/A
GS-200	60.16821621	36.35639754	41.49914004	24.13683875	68122.32117	340.6116059	51
400-Period					139299.6885	348.2492211	N/A
GS-400	58.55154955	33.51354007	41.02892651	28.76484078	136757.8517	341.8946292	70
1000-Period					352854.5207	352.8545207	N/A
GS-1000	57.55154955	36.41354007	43.03802657	24.42255303	347967.6985	347.9676985	8
ROIP-10	55.55154955	33.51354007	43.03802657	24.13683875			
200-Period					70771.98228	353.8599114	N/A
GS-200	60.16821621	36.35639754	41.49914004	24.13683875	68122.32117	340.6116059	51
400-Period					139299.6885	348.2492211	N/A
GS-400	58.55154955	33.51354007	41.02892651	28.76484078	136757.8517	341.8946292	70
1000-Period					352854.5207	352.8545207	N/A
GS-1000	57.55154955	36.41354007	43.03802657	24.42255303	347967.6985	347.9676985	8
ROIP-20	58.13421276	30.93087685	43.03802657	20.87725288			
200-Period					71996.31483	359.9815741	N/A
GS-200	60.13421276	36.71381577	43.03802657	22.31725267	68208.76041	341.0438021	57
400-Period					141174.6484	352.936621	N/A
GS-400	58.13421276	34.26421019	42.30642002	27.43035648	136962.6112	342.4065279	114
1000-Period					357290.9984	357.2909984	N/A
GS-1000	58.13421276	36.11659114	43.03802657	24.24862467	347934.199	347.934199	41
ROIP-30	58.13421276	41.50990002	52.28296631	41.74361298			
200-Period					74837.21764	374.1860882	N/A
GS-200	57.13421276	38.84323335	41.13021597	41.74361298	69858.39418	349.2919709	928
400-Period					152091.8763	380.2296907	N/A
GS-400	56.13421276	36.63079497	44.40396182	41.74361298	141024.2766	352.5606915	900
1000-Period					382425.2248	382.4252248	N/A
GS-1000	57.46754609	37.60990002	42.10939067	41.74361298	355504.2008	355.5042008	892
ROIP-50	61.49077212	44.02684049	37.93301773	35.34849772			
200-Period					70800.11337	354.0005668	N/A
GS-200	59.01180387	39.49482745	34.96395886	35.34849772	68064.52587	340.3226293	83
400-Period					144117.4082	360.2935206	N/A
GS-400	57.87172451	35.38941608	37.93301773	35.34849772	136594.2587	341.4856467	258
1000-Period					362205.3326	362.2053326	N/A
GS-1000	57.24756676	38.03336557	36.88375418	35.34849772	346456.7405	346.4567405	159

Table A.50: Beta(2, 5, 32, 12): Experiment Run 10

Method	S1	S2	S3	S4	Optimal Cost	Optimal Cost Per Period	# Iterations
ROIP-0	0	0	0	0			
200-Period					940828.5511	4704.142755	N/A
GS-200	16.07253303	0	0	0	825684.9244	4128.424622	1736
400-Period					1907588.575	4768.971439	N/A
GS-400	16.0458505	0	0	0	1677105.979	4192.764947	1713
1000-Period					4794858.006	4794.858006	N/A
GS-1000	16.03765856	0	0	0	4218079.653	4218.079653	1706
ROIP-5	48.17379783	26.33365363	38.89658134	31.38803518			
200-Period					91205.36171	456.0268086	N/A
GS-200	61.86831507	27.16154092	38.89658134	31.38803518	77320.08733	386.6004366	800
400-Period					183106.64	457.7666	N/A
GS-400	59.53405338	29.96620881	38.89658134	31.38803518	148546.8324	371.3670809	1012
1000-Period					448360.1128	448.3601128	N/A
GS-1000	61.42060852	28.24285463	38.89658134	31.38803518	358678.9751	358.6789751	1098
ROIP-10	51.06257523	23.44487624	38.89658134	31.38803518			
200-Period					90589.67435	452.9483718	N/A
GS-200	61.92886941	27.10098658	38.89658134	31.38803518	77319.17901	386.5958951	800
400-Period					182385.0496	455.9626241	N/A
GS-400	59.71606084	29.78024291	38.89658134	31.38803518	148554.5946	371.3864864	1010
1000-Period					445927.4425	445.9274425	N/A
GS-1000	61.9597579	27.67802861	38.89658134	31.38803518	358762.2624	358.7622624	1084
ROIP-20	51.06257523	23.42879363	38.91266395	31.37195257			
200-Period					90615.53519	453.077676	N/A
GS-200	61.94437329	27.06940009	38.91266395	31.37195257	77323.93207	386.6196603	800
400-Period					182439.3928	456.0984819	N/A
GS-400	59.71606084	29.7641603	38.91266395	31.37195257	148567.332	371.41833	1010
1000-Period					446063.0028	446.0630028	N/A
GS-1000	61.9597579	27.66194599	38.91266395	31.37195257	358791.8866	358.7918866	1084
ROIP-30	51.09256632	43.79421117	43.79421117	27.51672688			
200-Period					77454.31401	387.27157	N/A
GS-200	55.58914907	42.44548212	40.48866218	27.51672688	76532.49572	382.6624786	55
400-Period					147461.4752	368.6536879	N/A
GS-400	55.86944128	39.02223206	43.79421117	27.51672688	145956.093	364.8902325	67
1000-Period					358473.5056	358.4735056	N/A
GS-1000	56.32479848	39.94631101	42.03942233	27.51672688	353389.1967	353.3891967	127
ROIP-50	51.16479394	43.72198355	27.98561736	43.32532069			
200-Period					77845.66686	389.2283343	N/A
GS-200	54.83146061	43.72198355	27.98561736	42.82532069	76648.94483	383.2447242	5
400-Period					148225.0277	370.5625692	N/A
GS-400	56.28359669	43.30307109	27.98561736	40.0296614	145634.7784	364.086946	47
1000-Period					360160.5152	360.1605152	N/A
GS-1000	56.05003321	43.72198355	27.98561736	38.63185514	352981.6141	352.9816141	68