Title
The Anti-Covering Location Problem: new modeling perspectives and solution approaches

Author
Niblett, Matthew Russell

Publication Date
2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Santa Barbara

The Anti-Covering Location Problem:

new modeling perspectives and solution approaches

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Geography

by

Matthew Russell Niblett

Committee in charge:

Professor Richard L. Church, Chair

Professor Keith C. Clarke

Professor Stuart H. Sweeney

Professor Anthony H. Grubesic, Drexel University

September 2014

The dissertation of Matthew Russell Niblett is approved.

Keith C. Clarke

Stuart H. Sweeney

Tony H. Grubesic

Richard L. Church, Committee Chair

September 2014

The Anti-Covering Location Problem:

new modeling perspectives and solution approaches

by

Matthew Russell Niblett

ACKNOWLEDGEMENTS

This manuscript is dedicated to those who have helped guide me in my academic and personal pursuits. In particular, I thank all of the teachers I was blessed to have had as a student; I hope that I can be as inspirational and effective in my teaching as you were to me. I thank all of my friends for being there when I needed to talk or to let loose. I thank my parents for supporting and encouraging me always, especially when I have felt defeated. I also thank Lacey for her encouragement and support and for understanding when I had to work long nights.

VITA OF MATTHEW RUSSELL NIBLETT
June 2014

EDUCATION

Bachelor of Science in Physical Geography, University of California, Santa Barbara, June 2006 (cum laude)
Master of Arts in Geography, University of California, Santa Barbara, March 2009
Doctor of Philosophy in Geography, University of California, Santa Barbara, September 2014

PROFESSIONAL EMPLOYMENT

2004-2005: Undergraduate Research Assistant, Department of Geography, University of California, Santa Barbara
Summer 2006: Summer Internship, Ventura County Resource Management Agency GIS Division
Fall 2006-Spring 2007: Teaching Assistant, Department of Geography, University of California, Santa Barbara
Summer 2007-Summer 2014: Graduate Research Assistant, University of California, Santa Barbara

PUBLICATIONS

"Optimal Scheduling of Forest Treatments to Reduce Fire Intensity," Unpublished thesis submitted in partial fulfillment of the requirements for the Master of Arts degree in Geography, University of California, Santa Barbara, 1996. 58 pp.

"Modeling the Potential for Critical Habitat," book chapter by Richard L. Church, Matthew R. Niblett, and Ross A. Gerrard, 2014.

"Structure of fisher (*Martes pennanti*) habitat in a managed forest in an interior Northern California coast range," Forest Science, *accepted with minor revision*.

"The disruptive anti-covering location problem," European Jouranal of Operational Research, *in review*.

AWARDS

Outstanding Achievement as a Geography Major, University of California, Santa Barbara, 2006

Jack and Laura Dangermond Travel Scholarship, 2012, 2013

FIELDS OF STUDY

Major Field: Geography

Studies in Operations Research and Location Analysis with Professor Richard L. Church and Tony H. Grubesic

Studies in Spatial Statistics and Location Analysis with Professor Stuart Sweeney

Studies in Geographic Information Science with Professor Keith Clarke

ABSTRACT

The Disruptive Anti-Covering Location Problem: new modeling perspectives and solution approaches

by

Matthew Russell Niblett

Dispersive strategies and outcomes are readily apparent in many geographic contexts. In particular, dispersive strategies can be seen in activities such as: the scattering of military missile silos and ammunition bunkers, center-pivot crop irrigation systems, location of parks, franchise store location, and territorial species den/nest locations. Spatial optimization models represent dispersion where selected facility locations are maximally "packed" or maximally "separated." The Anti-Covering Location Problem, in particular, is one in which a maximum number of facilities are located within a region such that each facility is separated by at least a minimum distance standard from all others. In this context, facilities are "dispersed" from each other through the use of the minimum separation standard. Solutions to this problem are called maximally "packed" as there exists no opportunity to add facilities without violating minimum separation standards.

The Anti-Covering Location Problem (ACLP) can be defined on a continuous space domain, or more commonly, using a finite set of discrete locations. In this dissertation, it is assumed that there exists a discrete set of sites, among which a number will be selected for facility locations, and that this general problem may represent a number of different problems ranging from habitat analysis to public policy analysis. The main objective of this dissertation is to propose a new and improved optimization model for the ACLP when applied to a discrete set of points on a Cartesian plane using a combination of separation conditions called core-and-wedge constraints. This model structure, by its very definition, demonstrates that all planar problems can be defined using at most seven clique constraints for each site. In addition, the use of an added set of facet constraints in reducing computational effort is explored.

Anti-covering location model solutions are maximally packed, providing an "optimistic" estimate of what may be possible in dispersing facilities. But, what if less than optimal sites are employed in a dispersive pattern. That is, to what extent can an optimal maximally packed configuration be disrupted? This possibility is explored through the development of a new model, called the Disruptive Anti-Covering location model.

TABLE OF CONTENTS

LIST OF FIGURES

I. Dispersion

A. Introduction

Dispersion is directly related to location problems. In such problems, dispersion between facilities, and or demands, has often been measured as a function of distance. When considering dispersive behavior, it is often within the context of facility site location. Facility site location has long been a topic of interest to researchers in geography, economics, engineering, and planning. To understand dispersion, one must first discuss facility site location.

Facility site location has been of importance to mankind since pre-historic times when “hunter and gatherers” chose sites for encampments. Such location decisions were tantamount to long term survival. This interest in site selection has never waned and has even intensified since the emergence of cities, agricultural practices, scientific principles, and industrial economies. For example in the 1600’s Pierre de Fermat described a point location problem as follows: “given three points on the plane, find a fourth point which minimizes the sum of the distances to the other three points.” Since these early beginnings, geographers, as well as regional scientists, industrial engineers, econometricians and business planners have formulated prescriptive models for facility location and general location questions.

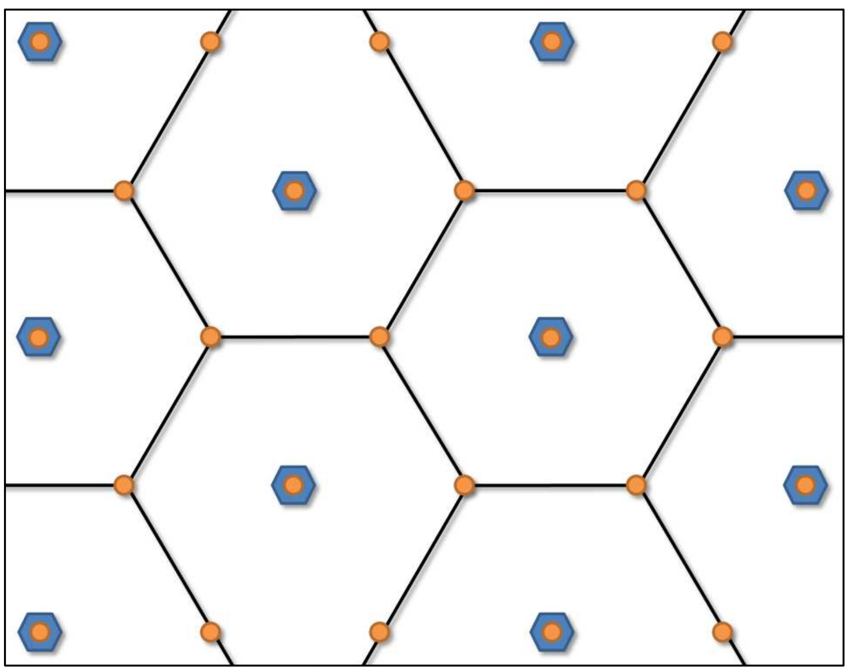
1. Location Theory

An early example of a quantitative description of how best to locate a factory is the classic Weber problem (Weber, 1909) that involves locating a manufacturing plant such that the costs of transporting the raw materials to the plant and the finished goods from the plant to the market are minimized. Another classic example is Christaller's (1933) central place theory, which theorizes an explanation of why retail centers were arranged in distinct patterns in southern Germany. Christaller assumed a competitive economic process of retail owners who located in such a manner as to achieve at least a minimum threshold of business activity, assuming that products and prices among retailers were indistinguishable.

Christaller extended the concepts of how distance and demand affected the placement of markets relative to an underlying base of customers. Christaller references earlier researchers, such as Hettner, who in 1902 had suggested areas of research that should be explored and which could quantify and explain, "Distances between settlements of the same economic character." Christaller and Hettner were both concerned with the location of facilities or types of spatial entities, and why they were uniquely dispersed across the landscape. Christaller believed that distance and the cost of obtaining resources of varying value are key elements that led to such a dispersive pattern. Christaller identified two aspects of spatial dispersion that are critical components of his theory; threshold and range.

The concepts of threshold and range in the central place model can be conceptualized in varying ways. In terms of business economics, the threshold can be thought of as the minimum level of demand needed to sustain a business facility at a given location, and the range is the maximum distance a person is willing to travel to buy a good or service (Hurst, 1972). Christaller recognized that there seemed to be a dispersed pattern of cities providing services to the surrounding places, and that the concepts of threshold and range are essential to understanding such dispersion of places. An example of a central place pattern for a market orientation is given in the Figure 1. Notice that central places are dispersed in a unique geometrical arrangement.

Figure 1. Maximally dispersed central place theory arrangement on 25 nodes $(K = 3)$



Church and Bell (1990) examined the spatial and economic implications of relaxing the geometrical packing requirement of classical Central Place Theory. They found that relaxing the geometric packing constraint still resulted in demand being satisfied. They found that with unpacked landscapes, those that resulted from relaxing the packing constraint, fewer facilities were required to meet demand and that often they were more widely spaced though did not increase the length of journey to a shop. Additionally, they found that stable *k*-systems emerged, that is, where no further market entry was possible just as in Christaller's classic work. Church and Bell further note that non-integer *k*-systems are possible, as are systems that are stable combinations of co-existing *k* principles. One critical point of this outcome is that consumers have fewer choices in an unpacked landscape, but on the upside, economies of scale could increase the array of goods and services available. Furthermore, they found that if two entrepreneurs co-located in the same central place rather than monopolizing a limited hinterland, the two would actually receive greater economic benefits. Church and Bell (1990) suggest this result is consistent with retail trends of the 1990's and the duplication ratio concept of Berry and Garrison (1958). Thus, Church and Bell have shown that dispersion between facilities is possible, and in many ways, desirable and more representative of the "real world." One interesting point that Church and Bell (1990) did not consider is whether such centers are maximally dispersed. However, Church and Bell (1990) mention a sequel to their paper dealing with demands distributed in a discrete, punctiform, manner that would require the same or fewer supply centers at any hierarchical level; it appears this paper was never developed.

It took seventeen years before research involving Central Place Theory within the context of whether central place facilities will result from a maximally dispersed process to be considered. In 2007, Curtin and Church developed a model for which facilities were maximally dispersed but had protected thresholds. They formulated two dispersion models: one that considers the single-good system, and one that deals with a multiple-good system. Specifically, two methods for generating multiple-good systems are presented: a multiple-type dispersion model and a *k*-value constraint set formulation. These formulations allowed the hierarchical systems to grow by increasing the number of maximally dispersed places. Their paper showed that stable k-levels were identified that met the classic tenets of central place theory for maximally dispersed facilities. They suggest that, “The objective of maximal dispersion is posited as both a motivating factor in central place location decisions, and as the optimal outcome of a mature system of central places (Curtin & Church, 2007, p.167).” In other words, the classic central place patterns of Christaller’s central place theory have been shown to be maximally dispersed (Curtin & Church, 2007).

Central place theory, however, is not only limited to cities and economics. One can observe a similar phenomenon, the location of dens/nesting sites, in territorial species. There are underlying factors that influence where a nest or den site is located and how the maintenance of territory effects the distribution of these nest/den sites across a landscape. In this case, the threshold can be thought of as the minimum level of available resources necessary for a territorial species to support themselves and maintain a nest/den site at a location, and the range can be thought of as the maximum energy expended or the furthest distance that an animal could reasonably travel in order to maintain a reliable supply of food. If a territory does not contain enough food resources within a reasonable reach to support an animal, it fails to meet a threshold of success, just as a retail site fails when the customer base falls short of a threshold within the range of the good. Just like central places, viable territories or “home ranges,” must be dispersed. In both the economic and territorial species conceptualization of central place theory, dispersive strategies are necessary to maintain a viable business location or maintenance of a territory.

2. Use of Dispersive Strategies

Dispersive strategies are observed in many different locations at varying scales. Dispersion manifests itself in the location of territorial species den/nesting site location, forest management activities, urban park locations, strategic facility placement, competitive retail store location[1], obnoxious facility location, halfway houses, and the location of correctional rehabilitation centers among others. In each of these examples there is an underlying process, strategy, or objective that tends to generate spatially dispersed activities.

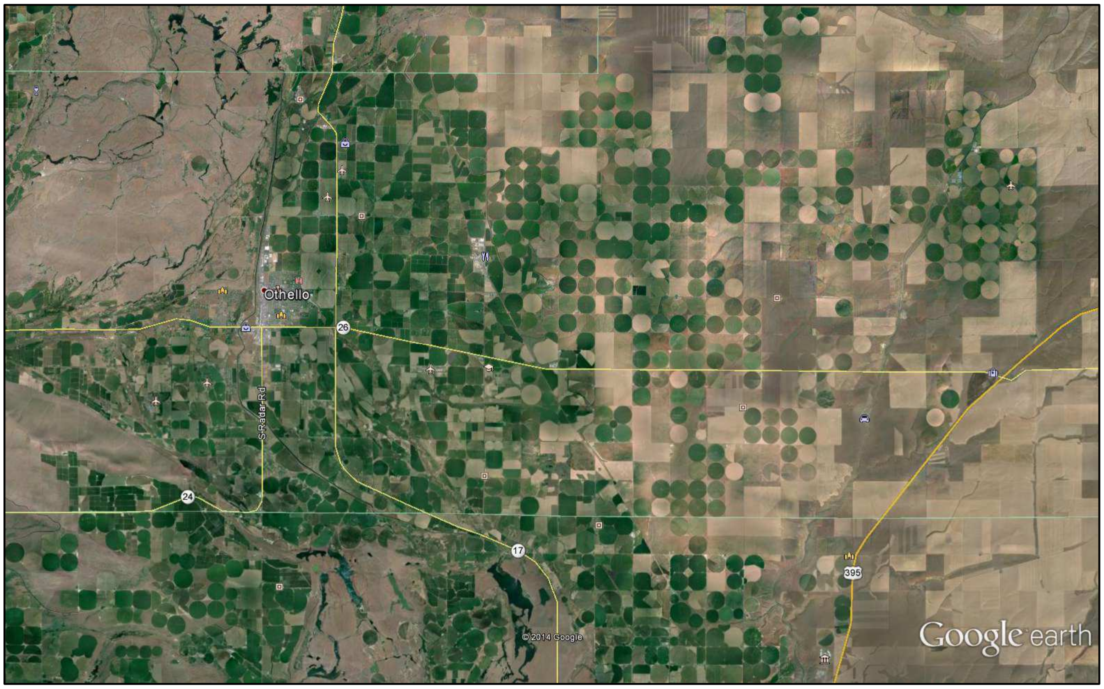
In the territorial species den/nest site case, dispersion is caused by the necessity of protecting the food source and to preserve a suitable site to secure the success of the species progeny. Similarly, with forest management activities, such as logging and fuels removal for the reduction of forest fire intensity, one wants to disperse these activities so that no one part of the forest is overly impacted by these activities.

[1] Competitive retail could be chain stores that sell goods, banking facilities, and franchise stores such as automobile sales lots and fast food restaurants.

Competitive retail store location, particularly for franchises, is similar to the territorial species case; the franchiser does not want too many franchises located close together because they will cannibalize sales from each other and hinder the success of an individual franchisee.

In the case of obnoxious facility location (e.g. landfills, half-way houses, and correctional rehabilitation centers) or semi-obnoxious facilities (e.g. fire stations, police precincts, etc.), policy makers try to spread these facilities out so as to minimize the impact to the public at large (Church & Garfinkel, 1978; Erkut & Neuman, 1989). In rehabilitation homes, in particular, dispersion is sought after to minimize the interaction of individuals with negative societal influences, and thus the recidivism rate of the individuals (Grubesic et al. 2011). Another example of strategic placement of facilities is the well of a center pivot irrigation system. Such center-pivot systems are often separated in such a way as to maximize irrigated areas as efficiently as possible with variable sized center pivot systems (New & Fipps, 2000). Figure 2 shows an example of a packed configuration of center-pivot irrigation systems in eastern Washington state, USA. The examples listed above are only a sampling; there are many examples of dispersive processes and locational outcomes in the context of ecology (Church, 2013), business economics (Erkut & Neuman, 1989), and social institutions (Grubesic et al. 2011).

Figure 2. Example of several center-pivot irrigation systems expanding in farmlands of eastern Washington state, USA. Center of image is: 46°48'16.98" N 119°00'31.84" W, from Google Earth



B. Modeling Dispersion

Dispersion has been an objective of considerable interest in the field of location science. There are three basic forms of dispersion. The first involves the dispersal of facilities from population centers (See Church & Cohon, 1976; Church & Garfinkel, 1978 as early examples of this type of problem). A second form of dispersion involves the dispersal of facilities from each other. Keeping facilities as far apart as possible from each other has been the subject in a number of different problem settings, ranging from military defense (Erkut, 1990) to franchisee store location (Current & Storbeck, 1994). A third form of dispersion, which is a hybrid of the first two forms, involves keeping facilities away from each other as well as away from population (Berman & Huang, 2008).

Moon and Chaudhry (1984) were among the first to propose a formal problem dispersing facilities from each other on a network, called the *p*-dispersion problem. This problem involves locating "*p*" facilities on a network, such that the minimum distance of separation between the closest pair of facilities is maximized. Moon and Chaudhry (1984) also proposed a model which maximized the sum of minimum separation distances, with one separation distance defined for each facility, while locating p-facilities. This problem was called the *p*-defense problem. Kuby (1987) expanded this concept to a problem that involves maximizing the sum of all separation distances between all pairs of facilities. Erkut and Neuman (1991) added a fourth classic form which involves locating p-facilities as well. For their problem, each facility is represented by the sum of separation distances to the other *p*-1 facilities. Their objective was to maximize the smallest of these facility defined sums. Curtin and Church (2006) proposed general forms of these problems which involve the location of different types of facilities, where interaction between different types has a defined repulsion weight and Lei and Church (2013) have shown that all four classic forms outlined by Erkut and Neuman (1991) can be viewed as special cases of general dispersion model using a concept based on vector assignment.

There is one other important form of facility dispersion and it is based upon a standard of minimum separation. Moon and Chaudhry (1984) were the first to focus on a minimum separation standard. They proposed to locate as many facilities as possible while keeping them at least *r*-distance apart from each other. They called this the anti-covering location problem. It has been used in a number of different ways. Grubesic and Murray (2008) proposed its use in analyzing policies that dictate that sex offender residences should be kept separated from each other as well as from selected fixed elements on the landscape, like parks and schools. Downs et al. (2008) used the anti-cover problem to analyze the carrying capacity of a population of Sandhill cranes, Williams (2008) employed a separation distance in the selection of biological reserve sites, Church (2013) has used it in estimating the size and extent of core habitat, and Murray and Church (1996) describe a form of anti-covering for a forest harvest selection problem. Grubesic et al. (2012) analyzed the impacts of alcohol outlet distribution in Philadelphia based upon a proposed policy change involving privatization.

More general forms of this problem have been defined for dashboard layout (Castillo et al. 2008), map label placement (Ribeiro & Lorena, 2008a), DNA sequencing (Joseph, Meidanis, & Tiwari, 1992) and the location of undesirable facilities (Berman & Huang, 2008). A number of techniques have been used to solve the anti-cover problem and related problems, including greedy (Chaudhry et al. 1986), bee colony optimization (Dimitrijević et al. 2012), Lagrangian relaxation (Murray & Church, 1997b), genetic algorithms (Chaudhry, 2006), column generation (Ribeiro & Lorena, 2008a), and greedy randomized adaptive search (Cravo et al. 2008).

Most of the applications of the anti-covering model entail the use of an integer-linear programming model. Prospective sites are often identified in advance as "discrete" locations, representing centers of raster cells (Church, 2013), commercial parcels (Grubesic et al. 2012), or nodes of a network. Murray and Church (1997a) have shown that the discrete anti-cover problem is an equivalent problem to the vertex packing problem on a network or the maximal independent set problem on a graph. This demonstrates that the discrete anti-cover problem belongs to the class of non-deterministic polynomial-time (NP) hard. There can be possible uncertainty in potential site positions, and Wei and Murray (2012) have analyzed the impacts of site uncertainty within the context of the anti-cover problem.

Research that is focused on modeling anti-covering is extensive and a number of model formulations have been proposed. In chapter three, these model structures are reviewed. After that a new form of ACLP is proposed based upon a new concept of "Core and Wedge." With this concept, it is shown that all Euclidean-based discrete point anti-covering models can be formulated with at most 7 clique constraints per site. Optimal solutions to the anti-covering problem represent the largest number of facilities that can be simultaneously located while keeping each of them at least a minimum distance, *r*, from each other. Unfortunately, there can be circumstances in which a maximum packing is disrupted. They may be disrupted by earlier residential choices, already established crane nests and territories, or by poor choices in already located franchisee establishments. Whether maximal packing arrangements are disrupted by accident, happenstance or by intent, such disruption and the potential impact of disruption should be of interest when using this type of model. In chapter three, a description of the two basic ways in which the anti-covering problem has been formulated as an integer programming problem is given.

Following this a problem and model formulation is presented which seeks to maximally disrupt potential solutions to the anti-covering problem; that is, a model that identifies the minimum packing configuration. In addition, a model is also formulated that allows one to determine if there are other packing configurations between the maximally packed configuration, the Anti-Covering Location Problem (ACLP), and the minimally packed configuration, the Disruptive Anti-Covering Location Problem (DACLP). Chapter five presents the computational experience of: the previous and new “Core & Wedge” formulations representing the ACLP, and the formulations representing the DACLP. In addition, a new heuristic designed to quickly solve the ACLP on very large datasets is described and an example of a solution is given. Chapter six contains some concluding remarks and directions for future research.

C. Outline of Dissertation

The focus of this dissertation is on dispersion and its use with respect to facility location, and in particular, how it can be modeled within the context of an integer programming problem considering a discrete set of facility locations. This chapter briefly describes the structural organization of the dissertation and gives a brief overview of dispersion. The subsequent chapters are as follows: Chapter 2 reviews existing dispersion models. Chapter 3 describes classic and new mathematical methods of modeling dispersion within the context of the Anti-Covering Location Problem (ACLP). Chapter 4 describes two new ACLP models. Chapter 5 explores how such problems might be solved. Chapter 6 demonstrates how a real world dispersion problem can be modeled and solved. Finally, Chapter 7 concludes the dissertation with a discussion of the theoretical underpinnings, uses, and implementations of the dispersive techniques described in this dissertation.

II. A review of dispersive location strategies

This chapter reviews the location modeling literature pertaining to dispersion. The primary purpose of this chapter is to discuss how dispersion has been modeled previously. Previous models of dispersion can be broken down into two types of dispersive modeling. The first type can be thought of as maximizing a measure of dispersion, or "dispersiveness," between facilities to facilities, or facilities to demands. The second type of dispersive modeling is standards based. In this case, a minimum separation standard is employed. The nuances between these two ways of mathematically representing dispersion will be discussed. Formulations for the problems discussed in this chapter are not provided here, though, chapter 3 presents formulations specifically related to the standards based Anti-Covering Location Problem (ACLP), the focus of this dissertation. Now, let us begin the discussion of general dispersion problems.

A. Maximizing "Dispersiveness"

As mentioned above, dispersive modeling has taken two different paths. The first is maximizing a measure of "dispersiveness" and the second is a standards-based approach. The case where facilities are separated from their demands, and/or each other, has often been modeled as the p-Maxian problem. The case where facilities are separated from only each other has been termed the p-Dispersion problem. First, the p-Maxian Problem is discussed. Subsequently a discussion of the p-Dispersion problem will be presented.

B. Dispersiveness and the p-Maxian

The Maximum Median, or Maxian, problem was first described, with a formulation representing it, by Church and Garfinkel (1978). It was further expanded by Chandrasekaran and Daughety (1981) as a multiple p-facility location problem. The Maxian problem is the antithesis of the median problem where the objective is to site a facility as far as possible from centers of population. Church and Garfinkel (1978) describe the basis of a finite optimal set for a Maxian problem on a network. The p-Maxian problem is defined as follows: simultaneously locate *p* points (facilities) as far from each other *and* a given set of nodes (demands). This is the first such problem that considered dispersion between located facilities and a set of demands.

The *p*-Maxian problem maximizes “dispersiveness” between facilities and demands, as a function of distance, without a minimum separation standard. Church and Garfinkel (1978) note that the one-facility *p*-Maxian mathematical objective function is identical to the absolute median (*p*-Median) objective function of Hakimi (1964), except that the objective sense of Hakimi’s formulation is to minimize. In Hakimi’s *p*-Median problem “dispersiveness” is not desired; the median distance of a facility location and assigned demands is minimized. This is contrary to the *p*-Maxian problem where separation between facilities and demands is also maximized. Church and Garfinkel note that without an objective to disperse *p*-facilities apart from each other the *p*-Maxian solution would involve co-locating all *p*-facilities at the optimal 1-Maxian point. However, there are several ways that "dispersiveness" can be measured between facilities in a configuration. In particular, this relates to the *p*-Dispersion problem discussed below.

C. Dispersiveness and p-Dispersion

Several derivations of separation distance measures have been presented and discussed in the literature. These implementations have been used in the *p*-Dispersion problem. The *p*-Dispersion problem, first described by Shier (1977), involves the location of *p*-facilities such that the minimum distance between the closest pair of facilities is maximized. Shier suggested that this problem was ideal for placing a fixed number of fire hydrants over a street network. Kuby (1987) further notes that the *p*-Dispersion model could be used to avoid cannibalization of market areas for franchise stores.

Erkut and Neuman (1991) describe four possible objectives when maximizing dispersion between facilities. The paper by Curtin and Church (2006) contains a succinct, easily understood, synopsis of Erkut and Neuman's work. The classic representations of "dispersiveness" described by Erkut and Neuman for facilities of a single type using distance as a metric are as follows:

1) The Max-Min-Min representation maximizes the dispersion of the *p* facilities to be located. Specifically, the objective of the model is to Maximize the Minimum separation distance Minimum (hence Max-Min-Min) of each facility to its closest neighbor. That is, the objective considers the smallest separation distance of each facility and its closest neighboring facility and maximizes the separation distance of this pair. This particular representation

of dispersiveness is the classic form of what Moon and Chaudhry (1984) call the *p*-dispersion problem. However, Shier (1977) initially described this formulation as the p+1-dispersion problem, the dual of the p-Center problem. Several formulations of this model type have been developed with example applications and procedures in the literature (Chandrasekaran & Daughety, 1981; Kuby, 1987).

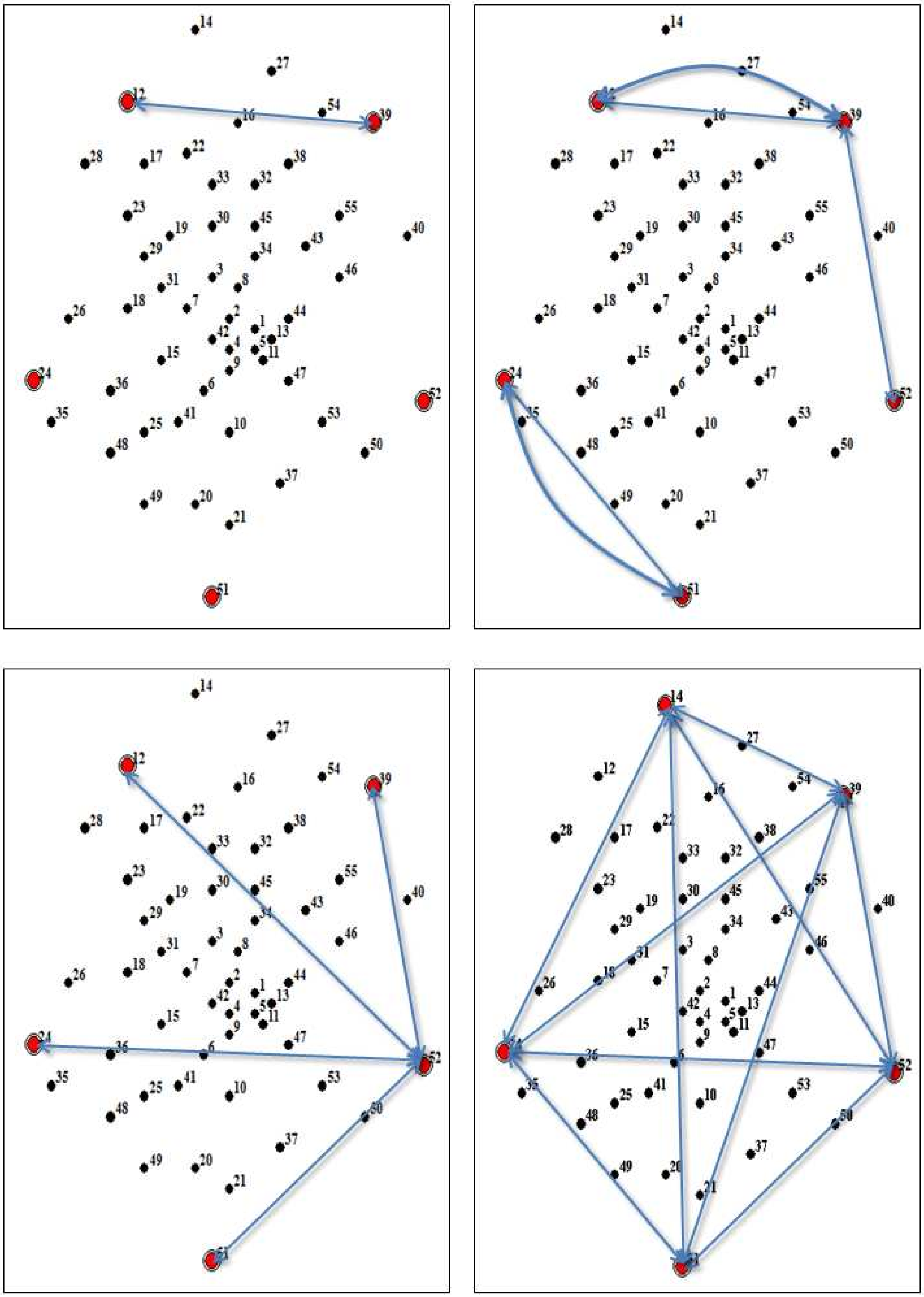
2) The Max-Sum-Min representation Maximizes the Sum of the Minimum (Max-Sum-Min) separation distances for every facility and the facility closest to it. This has been referred to as the *p*-Defense problem by Moon and Chaudhry (1984). The difference between the Max-Min-Min and the Max-Sum-Min is that the Max-Min-Min is concerned with the “worst case” of separation between a facility and its closest neighboring facility. The Max-Sum-Min is concerned with the *overall* sum of minimum separations between each facility and its closest neighboring facility.

3) The Max-Min-Sum representation Maximizes the Minimum of the Sum (Max-Min-Sum) of separation distances between a facility and all other facilities. Note the nuance here as compared to that found in the max-sum-min representation above. In this case, the separation distances of *all* facilities is considered, whereas in the max-sum-min representation, *only* the sum of the smallest separation distances between a facility and its closest facility is considered. This representation was first defined in Erkut and Neuman (1991).

4) The Max-Sum-Sum representation Maximizes the Sum of the Sum distances from a facility to all other facilities. Kuby (1987) formulated this problem, which he calls the Maxisum problem, that locates *p*-facilities *maxi*mizing the *sum* (Maxisum) distance between all pairs of open facility sites. Kuby's maxisum formulation locates facilities over a network or a set of discrete points. However, some facilities may be in close proximity to one another as the sum distance between *open* facility sites is maximized.

The four measures of "dispersiveness" presented above capture the four principal ways one might wish to separate facilities of the same type. Figure 3 shows these representations of dispersion graphically by locating five facilities, shown as the red dots. The top left panel shows the results of using the Max-Min-Min model, which maximizes the separation distance of the closest pair of facilities. This is indicated by the blue arrow connecting the closest pair of facilities. The top right panel shows the Max-Sum-Min model, where the sum of the distances, one for each facility to its closest neighboring facility, is maximized. These distances are shown by the blue arrows. The bottom left panel shows the results of the Max-Min-Sum model, where the minimum sum of separation distances from a facility to all other facilities is maximized. The blue arrows show which distances are accounted for in the Min-Sum (associated with site 52). The bottom right panel shows the solution of the Max-Sum-Sum model where the sum of all separation distances between located facilities is maximized, as indicated by the blue arrows. Erkut and Neuman point out that

Figure 3. Modeling outcomes of the Max-Min-Min (top left), Max-Sum-Min (top right), Max-Min-Sum (bottom left), and Max-Sum-Sum (bottom right) approaches to dispersion.



these four forms of dispersiveness can be grouped into two bases. The first base involves those objectives concerned with the “worst-case” separation distance between a facility and its closest neighbor, or with the sum of worst-case separation distance between all facilities. That is, “worst-case” separation efficiency versus collective sum of worst-cases, one for each facility. The second measure involves the *total* or *sum* of separation between facilities. Erkut and Neuman (1991) were the first to summarize the four basic models of dispersion.

Curtin and Church (2006) identified a nuance that the previous four models fail to account for: the concept of dispersion between facilities of differing types. They develop a general class of facility location models that optimize multiple type facility dispersion. This type of modeling has been successfully used to show that central places are maximally dispersed, and is an optimal outcome for which central places are mature (Curtin & Church, 2007). Additionally, they suggest strategies for constraint elimination to reduce computation time. Curtin and Church (2006) address dispersion between facilities of differing types and developed a form for each of the objectives codified by Erkut and Neuman (1991)

Lei and Church (2013) identified a concept that can be used to unify all four previous modeling implementations of dispersion. Lei and Church (2013) identified four new partial sum models that are generalized forms of the four models discussed by Erkut and Neuman (1991). The partial sum dispersion problem conceptually is quite simple. How dispersion is measured may be of greater importance to facilities that are closest to one another. For example, dispersing the closest 3 or 4 neighboring facilities to any given facility may only be of real interest, rather than only the closest facility or all facilities. Their paper models this type of dispersion using partial sums. Furthermore, they introduce a general model called Max-PSum-PSum which is a form which represents all basic forms of Erkut and Neuman (1991) as special cases.

In all of the representations of dispersion discussed to this point, no minimum separation standard has been considered. In addition to the previously discussed problems, there has been interest in formulating a model that will disperse facilities over a landscape based upon “equity” (Prokopyev, Kong, & Martinez-Torres, 2009). Such representations of equity could be seen as more easily implemented using a standard of separation. The next section focuses specifically on standards based dispersion.

1. Standards Based “Dispersiveness”

Dispersiveness can also be implemented through a standards based approach, typically implemented using distance or weighted distance where facilities are located where they have to be separated by at least a minimum distance of separation. Moon and Chaudhry (1984) introduced this construct in their seminal work of 1984. Moon and Chaudhry (1984) also proposed a classification scheme to introduce and define a variety of distance constrained problems. One of those problems is the Anti-Covering Location Problem, or *r*-Sep problem, that involves maximizing the number of facilities placed in a bounded region or problem domain, such that each facility is at least *r*-distance from each and every other located facility. This problem has also been called the packing problem in the field of mathematics and computer science, in which a maximal set of circles, spheres, or other polygons that can be tessellated, are packed into a finite 2 or 3 dimensional space. An introductory book describing various circle packing configurations within the context of mathematics has been written by Kenneth Stephenson (2005).

Conceptually the *p*-dispersion and *r*-separation, or anti-covering location problems are "duals" of each other. The p-Dispersion problem involves locating a fixed number of facilities and maximizing the separation distance, whereas the Anti-Covering Location Problem (ACLP) involves maximizing the number of facilities with a minimum separation distance standard. Even though these two problems can be considered duals of one another, past work has differed in their underlying formulations.

Yoshimoto and Brodie (1994) describe an approach that significantly tightens the neighborhood constraint used in the original formulation of Moon and Chaudhry (1984). Murray and Church (1996) developed an approach that further tightens the neighborhood constraint described by Yoshimoto and Brodie (1994). In addition, Murray and Church (1996) describe how cliques may be used to reduce several individual constraints into a single tight clique constraint. Erkut, ReVelle, and Ulkusal (1996) focused on generating integer friendly formulations of the Anti-Covering Location Problem (ACLP).

Their paper described six different formulations representing the ACLP. Their paper demonstrated the usefulness of varying formulation approaches that aid in obtaining integer solutions to the ACLP. The reason for doing this is that the ACLP is a difficult problem to solve to provable optimality, as it belongs to the NP-Hard[2] class of problems (Murray & Church, 1997a). Erkut et al. (1996) describe a particularly important formulation. They describe a formulation that utilizes the concept of neighborhood constraints and "clique" constraints. Specifically, Erkut et al. (1996) defined a clique constraint as containing the neighbors, *j*, of a potential facility location, *i*, that are all mutually within half of the separation standard of *i*, and that are within the separation standard of each other. This has been termed the "Core Clique" representation. A detailed description of this constraint approach is given in chapter three.

Murray and Church (1997a) describe a formulation to represent the ACLP which consists of a neighborhood constraint and a maximal clique. A maximal clique is defined as a set of neighbors *j* that are within the separation standard of site location *i* and all other members *j*. The neighborhood constraint contains all of the sites within the separation standard of *i* that are not members of the maximal clique. Murray and Church (1997a) show that the maximal clique can be computed by solving several vertex packing problems for all potential site locations. Several other applied papers have been written that involve the ACLP, that focus on several applications and problem solving approaches (Castillo et al. 2008). Castillo et al. (2008) discuss several packing problems, including one of packing various sized circles into a finite region, though they do not consider site-benefit in their formulation.

[2] NP-Hard stands for Non-deterministic Polynomial-time hard.

The Anti-Covering Location Problem has been used in modeling a broad spectrum of problems. For example, the ACLP has been used in: the Cartographic Label Placement Problem (Ribeiro & Lorena, 2008a); estimating carrying capacity of territorial species (Downs et al. 2008); evaluating planning policy and sex offender residency (Grubesic & Murray, 2008; Grubesic et al. 2011; Grubesic, et al. 2008); evaluating liquor store permitting and placement (Grubesic et al. 2012); DNA sequencing (Joseph, Meidanis, & Tiwari, 1992); forest planning problems (Murray & Church, 1995); center pivot irrigation systems (New & Fipps, 2000); analyzing historical and modern settlement patterns (Ruggles & Church, 1996; Curtin & Church, 2007); industrial problems such as container loading, dash-board layout, cutting patterns (Castillo et al. 2008) and fabric cutting patterns (Wong & Leung, 2009).

Approaches to solving these problems have varied; some problems are small enough that they are easily solved to optimality. When problems are not solvable to optimality, often another approach is required. Heuristics designed to generate feasible solutions as close to optimal as possible have been developed. Heuristic techniques have very different solution approaches. Such heuristic approaches include: greedy randomized adaptive search procedure or GRASP (Feo et al. 1994), genetic algorithms (Chaudhry, 2006), LaGrangian relaxation (Murray & Church, 1997b; Ribeiro & Lorena, 2008a; 2008b), tabu search (Yamamoto, Camara, & Lorena, 2002), Bee Colony Optimization meta-heuristics (Dimitrijević et al. 2012), and evolutionary algorithms (Wei & Murray, 2014). In the next chapter the Anti-

Covering Location Problem (ACLP) formulation is presented, followed by several refinements published in the literature. A discussion related to each formulated model and subsequent method of mathematical represented will be provided.

III. Mathematical representation of "Anti-Covering"

This chapter presents several mathematical models which represent the Anti-Covering Location Problem (ACLP). The chapter contains several sub-sections. The first section presents and discusses the classic ACLP model formulation developed by Moon and Chaudhry (1984). Subsequently, several concepts in constraint modeling are presented. They deal with: the concept of neighborhoods and their refinement, clique constraints, the concept of a "Core" constraint set, and hybrid approaches of the previous forms. Two completely new representations are also provided in this section: the concept of "Core and Wedge" cliques, as well as the use of location set covering (LSC) constraints in anti-cover modeling.

A brief discussion related to these new constraint methods and why and how one should implement them is also provided. In the next chapter, solution times and results related to each formulation are presented with a discussion on the strengths and weaknesses of each approach. This chapter presents each formulation and provides a brief description of how the constraints work, why the approach was developed, and some information related to ease of solving.

A. The classic Moon and Chaudhry (1984) Anti-Covering model formulation

As noted in the previous chapter, the anti-covering model was first proposed in Moon and Chaudhry (1984). The model that Moon and Chaudhry formulate is designed to solve the Anti-Covering Location Problem (ACLP), also known as the *r*-

Sep or radius of separation problem. The Anti-Covering Location Problem (ACLP) involves maximizing the number of facilities packed within a bounded region such that each facility meets a minimum separation standard, often times a distance measure such as *r*, from its closest located facility. In this section details in formulating the anti-cover location model (ACLM) is presented.

The anti-covering location model (ACLM1) developed by Moon and Chaudhry (1984) representing the discrete case is given as follows:

Notation:

i, j	are indices of potential facility locations
r	is the minimum distance standard, or radius of separation
S	is the set of potential facility site locations
Q_i	$Q_i = \{ j \in S \mid d_{ij} < r \text{ where } j \neq i \}$, defined for each $i \in S$
M	is a very large number (*i.e.* a big *M* value), at least equal to *n*, where *n* is the number of sites or $\lvert S \rvert$
d_{ij}	shortest distance from facility *i* to facility *j*
x_j	$= \begin{cases} 1, \text{if facility is sited at } j \\ 0, \text{otherwise} \end{cases}$

ACLM1:

$$\text{Maximize Z} = \sum_{j \in S} x_j \qquad (1)$$

s.t.

$$M(1 - x_i) \geq \sum_{j \in Q_i} x_j \qquad \text{for all } i \in S \qquad (2)$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in S \tag{3}$$

The formulation presented here is a zero-one linear programming problem comprised of $|S|$ binary variables and $|S|$ constraints. The objective of the ACLM1 (1) involves maximizing the number of sites selected to locate a facility. Constraints (2) ensure that if a given site *i* is selected for a facility, no other sites closer than *r*-distance of site *i* may be used. In effect, if $x_i = 1$, then the left hand side of the inequality must be zero in value. When the left hand side of this constraint is zero, the right hand side must also be zero. Essentially, this means that when the right hand side is forced to be zero, all facility sites within the set Q_i are unable to be selected as a facility. Constraints of type (3) represent the binary integer restrictions for the facility site selection variables. The total number of constraints, those that define the model, in this case are constraints of type (2). The binary integer restrictions (3) are not considered model constraints per-se; they represent the restrictions on the values a decision variable can take and are not a part of the constraint matrix. Hence, the total number of constraints found in the ACLM1 is equal to at most the number of facility sites ($|S|$ or $\leq n$ constraints). The number of constraints could be less because some facilities may not be within *r*-distance of another facility, and thus a constraint of type (2) would not need to be written.

Yoshimoto and Brodie (1994) formulated a model containing a similar constraint of type (2) above that was implemented for a forestry adjacency restriction problem, with an important difference. Yoshimoto and Brodie recognized that an arbitrarily large "big M" is not required when you know how many neighbors are within r of a facility. Specifically, the M in each constraint of type (2) can be replaced by the size of the set Q_i. Note that i is not a member of its neighborhood set, so the value of $|Q_i|$ equals the number of neighboring sites to site i. Moving from a large M value to a much smaller value such as $|Q_i|$ can greatly reduce the computational time required to solve a given problem to optimality because it is a tighter constraint. Consider, for example, the following equation: $x_1 + x_2 + x_3 \leq n_i x_i$. If n_i represents a large value, then variable x_1, x_2, and x_3 can be 1 in value when the value of x_i is a small fraction. This type of property means that constraints (2) can be easily violated when M is large and x_i is fractional. Often such constraints are not "enforced" without resolving fractional variable values with a branch and bound algorithm. If $|Q_i|$ is considerably smaller than M, then constraints (2) will be tighter and often require less effort in solving with branch and bound. Thus, any way to reduce the size of M in these types of constraints will generally reduce overall computation time necessary to solve a problem.

Murray and Church (1995) discuss how a big "M" can be further reduced by replacing M with the value n_i where:

n_i = the largest number of sites which can be simultaneously selected within the set Q_i while maintaining a distance separation of r between each pair of facilities (note $n_i \leq n$).

Murray and Church (1995) show that even though computing n_i requires solving a small vertex packing problem for each set Q_i, it has been shown that this can help to lower overall computational time. It is important to note that vertex packing problems are of the class of NP-Hard, so that large problems are challenges in themselves to solve. Where the Murray and Church (1995) approach makes sense is when each set Q_i is relatively small.

A second model form, the Anti-Covering Location Model 2 (ACLM2), first proposed by Erkut et al. (1996) and Murray and Church (1996) uses a constraint structure originally proposed by Thompson et al. (1973) that has been used to solve the ACLP. Using the previously defined notation, the model can be formulated as:

ACLM2

$$\text{Maximize } Z = \sum_{j \in S} x_j \tag{4}$$

s.t.

$$x_i + x_j \leq 1 \qquad \text{for each } i, j \in S \text{ where } i \neq j \text{ and } d_{ij} < r \tag{5}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in S \tag{6}$$

This second formulation of the ACLP has the same objective value (4) as that of the ACLM1, but uses what are called pairwise adjacency constraints (5). For each facility site pair, i and j where $i \neq j$, that are within r distance of each other, a constraint of type (5) is written. This constraint prevents more than one facility in the pair from being selected. Either facility *i* or facility *j* can be selected, or neither one when the model is solved. Constraints of type (6) are the binary integer restrictions for the facility site selection variable x_j. Whereas the ACLM1 is compact (having at most *n* constraints), the ACLM2 is not; this is due to the pairwise constraints of type (5). The number of constraints in the ACLM2 model is equivalent to the number of unique site pairs that are strictly within *r*-distance of each other, potentially a significant number. Murray and Church recognized that this constraint form can be reduced using higher ordered clique constraints. The next section discusses the concept of cliques in further detail.

B. Cliques

The Anti-Covering Location Model 2 (ACLM2) using pairwise constraints, see constraints (5) in previous section, can be reduced in number and tightened through the use of higher ordered cliques. To understand the representation of a higher ordered clique, let us consider three facility sites *t*, *u*, and *v* which are within *r*-distance of each other. The ACLM2 model formulation would contain the following pairwise constraints: $x_t + x_u \le 1$, $x_u + x_v \le 1$ and $x_t + x_v \le 1$. These three constraints can be represented, or reduced, into one inequality term: $x_t + x_u + x_v \le 1$, a clique constraint of 3 members. The model can be reduced by replacing these three pairwise constraints by this clique constraint. Clique constraints can be written when a set of sites are all mutually adjacent or within *r*-distance of each other.

It makes great sense to combine pairwise constraints whenever possible into higher ordered clique sets as this reduces the needed number of constraints and produces a tighter relaxed problem (Meneghin, Kirby, & Jones, 1988). In general a clique constraint can be written as: $\sum_{j \in C_k} x_j \leq 1$ where k is an index of clique sets K[3], and C_k is the set of members of cliques k, where each member of the clique is within r-distance of all other members of the sites within r of i. Murray and Church (1996) describe a method to determine the minimum number of clique sets that represent all pairwise constraints of a given problem. The Anti-Cover Location Problem (ACLP) as represented using ACLM2 with higher ordered cliques would use the following constraint:

$$\sum_{j \in C_k} x_j \leq 1 \qquad \text{for each } k \in K \tag{7}$$

instead of constraints of type (5). Through the use of cliques, the number of constraints found in pairwise formulations such as ACLM2 can be substantially reduced resulting in a more compact model that can be solved much faster. Cliques can also be implemented in several special ways. One such way is through the concept of a "Core" clique constraint set, discussed in the next section.

[3] K represents the set of all cliques

C. “Core” Clique Constraint Set Representation

“Core” sites are those sites that can be thought of as being so close to a particular facility, within $\frac{r}{2}$, that if a facility is located at any of the facilities within the core, no other facility can be placed. This property holds for problems defined on a Euclidean plane or problems in which the triangle inequality holds. Erkut et al. (1996) first described this type of constraint set within a model they call “Model IV”. A version of Erkut et al.’s model using the previously defined notation, formulated as the Anti-Covering Location Model – Core Clique Constraints (ACLM-CCC) here, is as follows.

ACLM-CCC

$$\text{Maximize } Z = \sum_{j \in S} x_j \tag{8}$$

s.t.

$$n_i x_i + \sum_{j \in K_i} x_j \leq n_i \qquad \text{for each facility } i \in S \tag{9}$$

$$x_i + \sum_{j \in CQ_i} x_j \leq 1 \qquad \text{for each facility } i \in S \tag{10}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in S \tag{11}$$

Where $CQ_i = \left\{ j \in S \mid d_{ij} \leq \frac{r}{2}, i \neq j \right\}$, defined for each $i \in S$ and

Where $K_i = \left\{ j \in S \mid \frac{r}{2} < d_{ij} < r, i \neq j \right\}$, defined for each $i \in S$

The objective of the ACLM-CCC (8) and the binary integer restrictions on the facility site selection variables (11) is the same as all of the other models formulated to this point. The significance of this model is in the implementation of constraints (9) and (10). Constraint (9) represents the form of constraint developed by Yoshimoto and Brodie (1994), which was previously discussed in relation to the ACLM1 model. As earlier noted, n_i represents the number of sites within Q_i. However, in this case, n_i can be reduced to represent the number of elements within K_i. Conceptually, constraint (9) specifies that if a facility at site *i* is selected, none of the other facilities strictly within *r* of site *i* can be selected. This represents the neighborhood constraint of facility *i*. If a facility at site *i* is not selected, than those facilities within *r* of site *i* remain candidates for selection. Constraint (10) is the "Core" constraint and can be thought of as a clique of facilities centered around facility *i* that are at least half-*r* away.

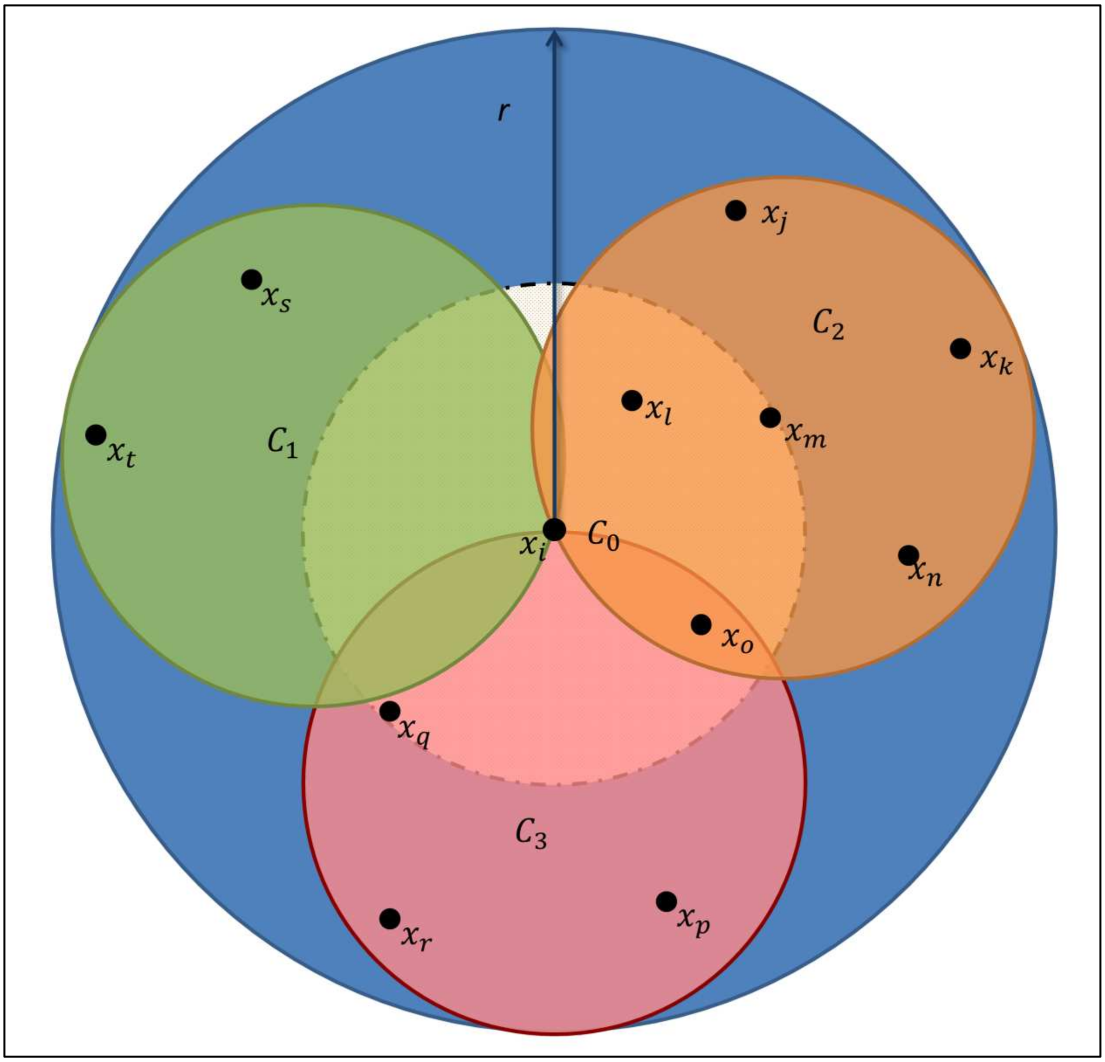
Constraint (10) represents a core area within $\frac{r}{2}$ distance of facility site *i*. If facility site locations *j* are strictly within $\frac{r}{2}$ of facility site *i*, any specific facility sites *j* within the core set of *i*, CQ_i, is also within *r* distance of all the other *j* facilities in the set CQ_i. This constraint is particularly tight as it requires that at most only one of the sites within the core can be selected at any time. As Erkut et al. (1996) note, even though constraint (9) sufficiently represents a feasible region, and that constraint (10) is redundant from a pure modeling point of view, constraint (10) tightens the problem considerably.

This is an important observation as the “Core” constraints represent a particularly tight neighborhood constraint about a facility which reduces the necessary computation time to solve the problem. Erkut et al. further note that the “Core” constraint set represents only a quarter of the total area within *r*-radius of site *i*, and that there could several potential facility locations not included within this type of clique constraint. However, constraint (10) is easy to compute and implement when building the location model. In fact, Erkut et al. also suggested an algorithmic way of further tightening the neighborhood constraints by eliminating members from the “core” set from the neighborhood set. However, there is a better method of generating constraints incorporating “Core” conditions than that suggested by Erkut et al. Murray and Church (1997a) proposed a “hybrid” form of a “Core” type clique constraint, called a Maximal Clique.

D. Hybrid Clique Constraint Representation

Murray and Church (1997a) describe how to generate a hybrid Anti-Covering Location Model (ACLM) through the use of a maximal clique set. The Anti-Covering Location Model – Core Constraints (ACLM-CC), described in the previous section, is in a certain way similar to the model that Murray and Church (1997a) describe. Murray and Church recognized that a clique constraint does not necessarily have to be centered on the location of facility *i*. Furthermore, they demonstrated that neighbors of facility site *i* can be grouped into a maximal set of neighbors that are all within r distance of one another, or a maximal clique. To conceptualize this, think of a small circle of radius half r located within a circle of radius r. You can move the circle of radius half r anywhere within the circle of radius r so long as you group the maximum number of potential facility locations within the half r circle. Note, that site i will always be a member of such a set. A maximal clique is conceptually similar to this example. A maximal clique can be <u>located anywhere within r of site i such that each of the other sites within r of i are also within r of each other</u>. Figure 4 contains a hypothetical set of sites within r distance of site i: sites j, k, l, m, n, o, p, q, r, s, and t. These facilities can be grouped into cliques, for example C_0, C_1, C_2, and C_3. C_0, the circle symbolized by the dash-dot pattern, represents the "Core" clique described by Erkut et al. (1996), discussed previously, which contains all facility sites $\leq \frac{r}{2}$ of site i. The "Core" clique constraint set, C_0, has 5 members: sites i, l, m, o, and q. The maximal clique, C_2 symbolized by the orange top-right circle, contains the greatest set of neighbors within r of each other; 7 in this case. The sites within the maximal clique are: i, j, k, l, m, n, and o. While the maximal clique set C_2 has been represented as a circle for demonstrative purposes, it need not necessarily be a circle. This is because maximal cliques are by definition determined through their mutual connections; that is, the maximal clique of facility site i and facility sites within r of i must also be within r of each other. Murray and Church (1997a) show that a constrained node packing problem can be used to compute maximal clique sets for use in solving the ACLP. This is done through the use of an undirected graph

Figure 4. Example of clique sets C_0, C_1, C_2, and C_3. The dashed circle with dot fill, C_0, represents the "Core" constraint clique of Erkut, ReVelle, & Ulkusal (r/2). C_2, the orange top right circle, represents the Maximal Clique for facility site *i*.



(network). Vertices (nodes) represent facility sites and the edges (arcs) represent pairs of sites that are < *r*.

To show how this looks in graphical form, let us re-use the example presented in Figure 4. Figure 5 shows this example graph. The small blue edges (arcs) represent theoretical connections to vertices (nodes) that are not within the *r*-neighborhood of site i. The edges (arcs) connecting to the other nodes represent those nodes that are also within r-distance of one another. For example clique C_1, sites i, s, and t, are represented on the graph as the orange edges (arcs) in the top left of Figure 5. Clique C_2 is represented as the brown arcs (top right), and clique C_3 is represented by the blue arcs (bottom left). The maximal clique, C_2, is determined by "packing" the number of nodes that are simultaneously connected together. However, as Murray and Church (1997a) have shown, maximal cliques for one facility site, say i, could be a subset of another maximal clique, say of site j. When a maximal clique for one facility site (i) is a subset of a maximal clique for another facility site (j) it is said to be dominated.

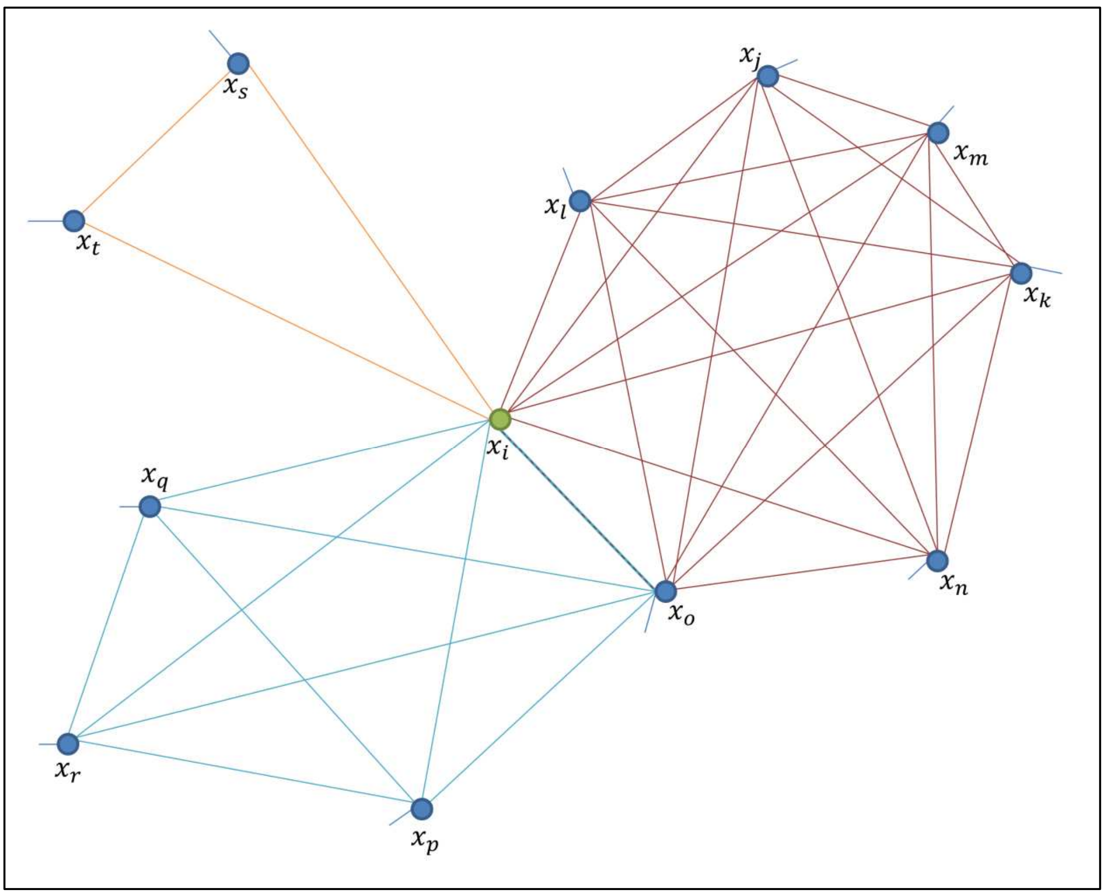
Murray and Church (1997a) thus suggest that the use of maximal non-dominated cliques should be used when formulating an ACLM. Fortunately, solving for non-dominated clique constraints is easily accomplished through industrial optimization packages or using the "back-tracking" method described by Nishizeki and Chiba (1988). Using maximal non-dominated cliques results in a compact formulation, consisting of tight clique constraints, which greatly reduce the necessary computation time to solve the ACLP to optimality. Murray and Church defined the following additional notation for the model using maximal non-dominated cliques, indexed by k.

$\hat{N}_i$ $= \{ j \in Q_i \mid i \,\&\, j \notin C_k \text{ for any other } k \in K \}$

$\hat{n}_i$ = Coefficient necessary to impose node packing restrictions for the set $\hat{N}_i$

$\hat{N}_i$ represents a reduced neighborhood set associated with site i, and $\hat{n}_i$ represents the coefficient necessary to impose node packing restrictions for the set $\hat{N}_i$. Essentially, all pairwise conditions handled in a clique constraint are removed from appropriate neighborhood sets. The maximal cliques are computed by solving a vertex packing problem for each candidate site location. This reduction allows for tighter (lower) values of $\hat{n}_i$ to be used. The maximal non-dominated clique model of

Figure 5. Cliques as a network representation.



Murray and Church, called the Anti-Cover Location Model-Hybrid Clique Constraints (ACLM-HCC) here, is provided below.

ACLM-HCC

$$\text{Maximize } Z = \sum_{j \in S} x_j \tag{12}$$

s.t.

$$\hat{n}_i x_i + \sum_{j \in \hat{N}_i} x_j \leq \hat{n}_i \qquad \text{for each } i\text{, where } \hat{N}_i \neq \emptyset \tag{13}$$

$$\sum_{i \in C_k} x_j \leq 1 \qquad \text{for each } k \in K \tag{14}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in S \tag{15}$$

The objective function (12) and the binary integer restrictions (15) are the same as used in previously described models. Constraint (13) represents those facility sites within *r* radius of facility site *i* that are not members of a non-dominated maximal clique; this represents a neighborhood constraint capturing all facility sites within *r* of *i* that are not members of a maximal non-dominated clique. This constraint is structured the same way as a general clique constraint; when x_i is 1, no other facilities within *r* of site *i* may be sited within that set. If x_i is 0, the other facility sites are still candidate sites. Constraint (14) represents maximal non-dominated cliques. At most one facility within each non-dominated clique may be located. This represents a very tight constraint while significantly reducing the overall size of the model. Such constraints significantly reduce the overall computation time necessary to solve the ACLP. However, the overall computation time may be greatly increased if one is using a dataset that contains a dense set of points due to the necessity of solving one vertex packing problem associated for each site in identifying the maximal clique set. Though this hybrid formulation is very efficient and significantly reduces required computation time, there is another approach that reduces the overall computation time even more than Murray and Church's hybrid approach. This approach is termed the "Core & Wedge" clique constraint approach, which was developed as a part of this dissertation research and is discussed in the next section.

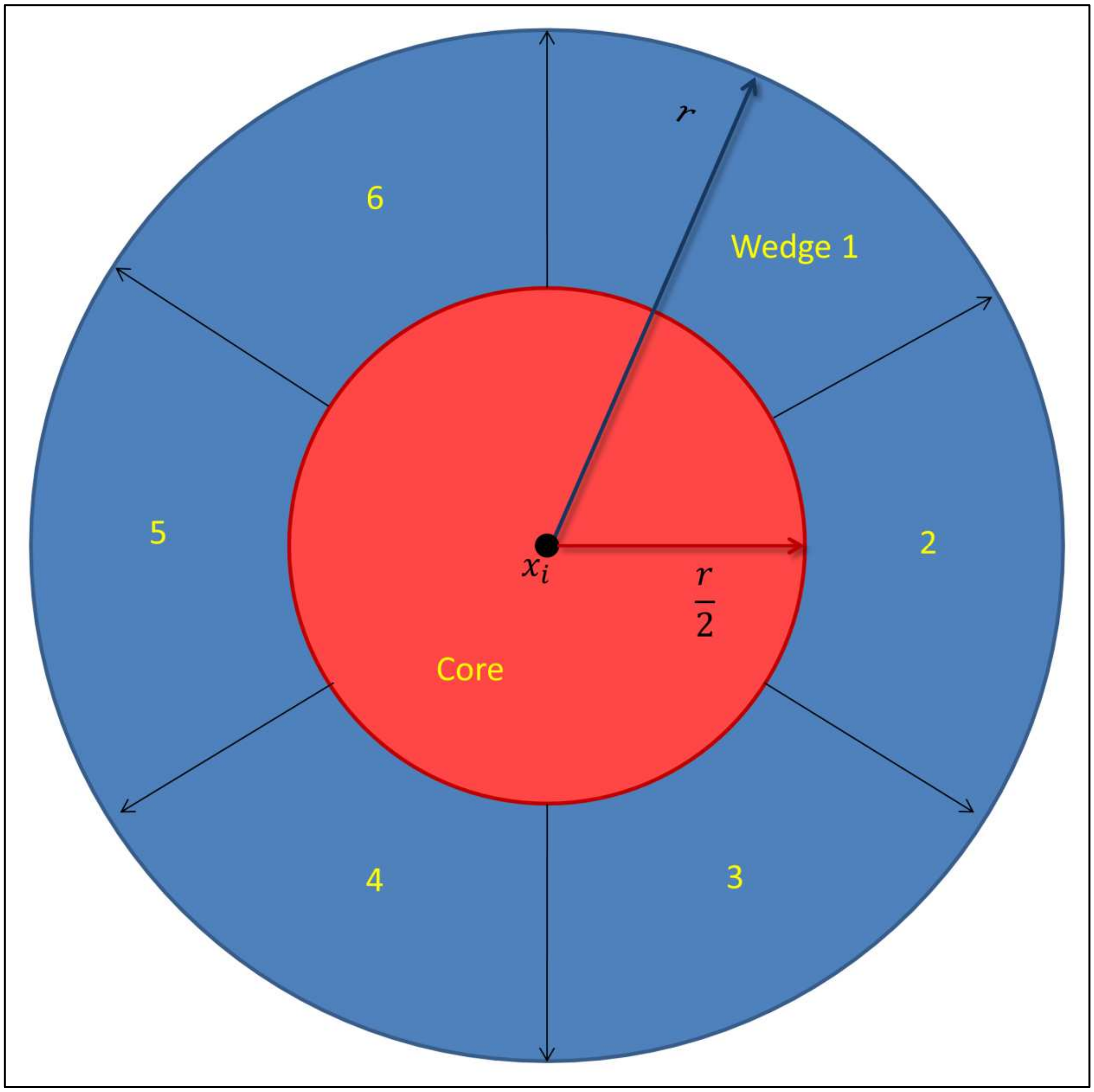
E. "Core and Wedge" Clique Constraint Representation

The "Core and Wedge" clique constraint representation uses the concept of the core clique constraint representation first described by Erkut et al. (1996) and the idea of off-center cliques (Maximal Cliques) first posited by Murray and Church (1997a). While Erkut et al. recognized a very simple way to easily create tight clique constraints, it still requires the relatively loose neighborhood constraint set. They failed to recognize that their method could be expanded further.

While Murray and Church recognized that one could group the set of facilities that are all within r of each other into a tight maximal clique constraint set, they failed to recognize that one could actually employ several sets of less dense clique sets. While their approach is effective at creating one very tight constraint for each site, which is particularly efficient for sparse data sets, it still requires the implementation of loose neighborhood constraints. Furthermore, their formulation approach requires numerous node-packing-problems to be solved just to identify the maximal non-dominated clique sets. That is to say, it can be computationally intensive to derive the maximal clique sets required to solve the ACLP-HCC model itself, particularly when the data set contains dense groups of points. The concept of "Core and Wedge" clique constraints is a very simple one. Every facility site location i has, at most, seven clique sets that capture all pairwise conditions within its neighborhood! A graphic conceptualization of "Core and Wedge" constraints is presented in Figure 6, 6, and 7.

The "Core and Wedge" diagram in Figure 6 is the result of a geometric analysis of the core circle and the larger region of points outside of the core, but inside the larger circle of radius r. Figure 6 depicts a site i, its core and the large circle of radius r. In addition, there is a wedge depicted which has been drawn with an angle of 60 degrees. If one thinks of the region outside the core, but inside of the larger circle as a tire, then a wedge is a region of that tire (or a felloe of a wagon wheel). If a wedge is defined to be equal to or less than 60 degrees, then all points within a wedge are within $r/2$ of each other. This can be proven by geometric construction and is depicted in Figure 7. This same wedge is shown with a circle of $r/2$ centered within the wedge itself. If a wedge has an angle of definition that exceeds 60 degrees then a circle of radius $r/2$ cannot be drawn that is centered within the wedge which covers all points within the wedge. Thus, any wedge (or felloe) that is defined that is less than 60 degrees contains a set of points that are close enough together, that they represent a core-like set. Consider then the following property:

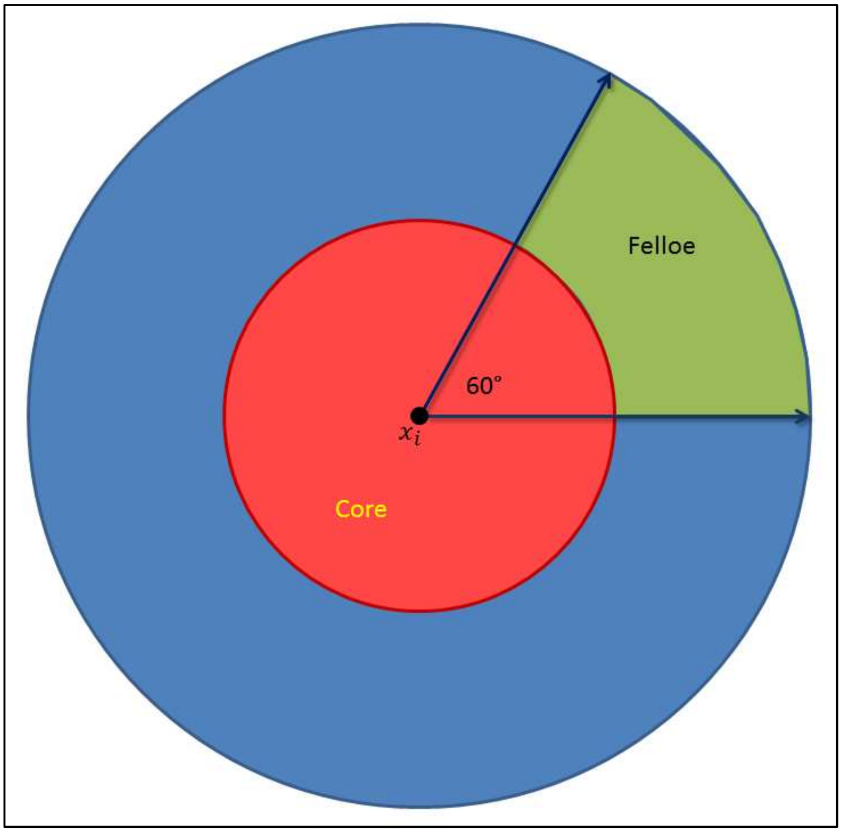
Figure 6. Example of Core and Wedge areas associated with site *i*



Corollary: A set of points that fall within a 60 degree wedge defined about point *i* along with point *i* form a clique set.

Proof: All points within the wedge set are within *r* distance of each other, so that the points within the wedge region about point *i* form a clique set. Since point *i* is also strictly within *r* distance of all points within the wedge, then point *i* can be added to the clique set. Thus, the points in the wedge set along with point *i* form a clique. QED

Figure 7. Example of a Core and Felloe region of a 60 degree angle

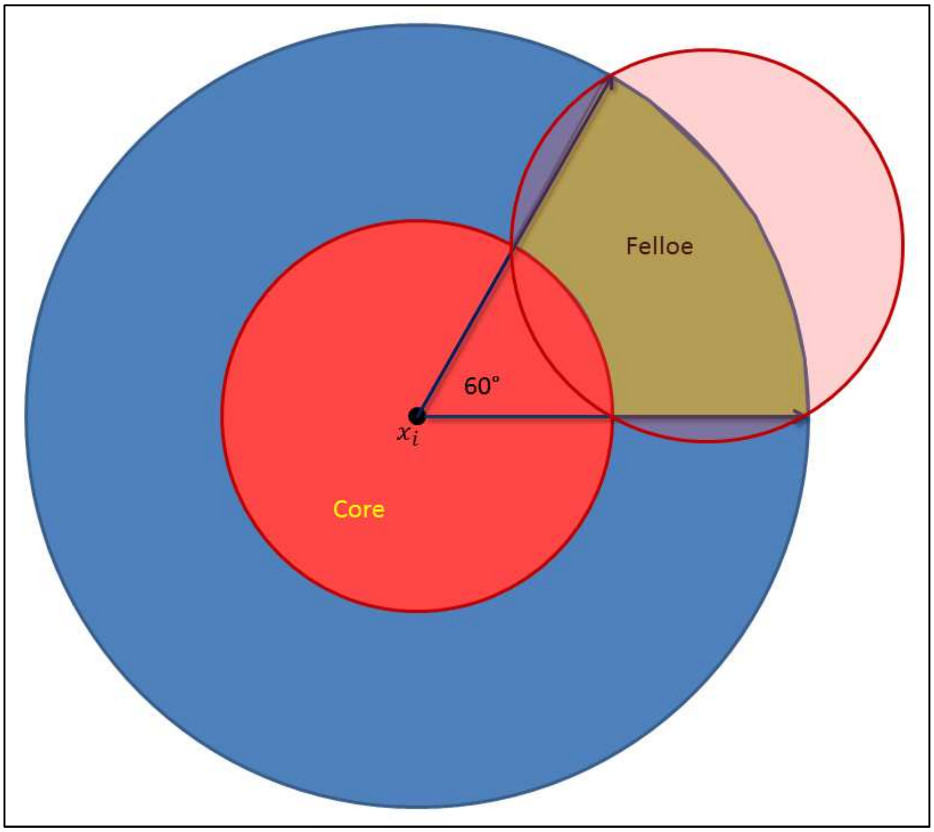


One can now define:

Wedge clique set is the set of points that fall within a 60 degree or less wedge of point *i* along with point *i*.

Since a circle is comprised of 360 degrees, the entire region within *r* distance about a given site *i*, can be represented by 6 wedge sets and a core set. Figure 8 depicts this construction. Note that the construction can be made where the first wedge drawn from point *i* within the circle of radius *r* can be oriented at any angle.

Figure 8. Example of felloe region with *r*/2 circle overlay



In Figure 6, the first wedge has been defined where one edge of the sector coincides with the vertical or *y*-axis. Now consider the following theorem.

Core and Wedge theorem: All pairwise restrictions associated with a given site *i*, can be represented with at most six wedge cliques and a core clique.

Proof: Each wedge set defines a clique set which contains point *i*, thus it represents all pairwise conditions associated with point *i* that fall within the wedge. Since all points strictly within *r* distance of point *i* will fall within at most 6 non-overlapping wedge sets or within its core set and since all of the pairwise restrictions within each of these sets can be represented by their associated clique set constraints, all pairwise restrictions associated with site *i* can be represented by at most 7 clique constraints. QED.

In Figure 8 the core constraint of Erkut et al. (1996) is shown as the red circle with radius $\frac{r}{2}$. Any facility site strictly within $\frac{r}{2}$ of facility *i* is a member of the core clique constraint. In addition to the "core" clique, six additional "wedge" clique constraints can also be constructed. In Figure 6 these are labeled in yellow print "Wedge 1", "2", "3", "4", "5", and "6". Consider the following notation:

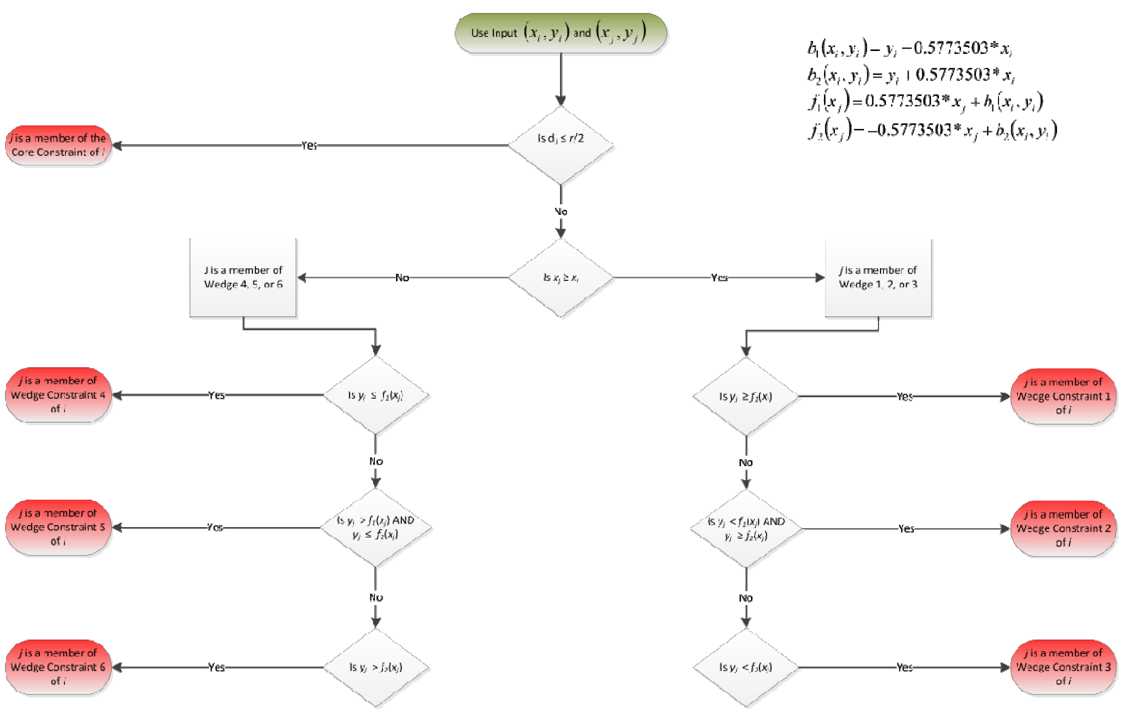
W_{wi} = Set of facility sites contained within wedge *w* of site *i*.

The wedge index ranges from one to six to account for each wedge. Computing W_{wi} is relatively easy and can be computed using a straightforward set of logical tests to determine which wedge set neighbor *j* within *r* of *i* should be assigned. When computing the "Core and Wedge" constraints for facility *i*, one need only work with the set Q_i to determine which core or wedge set facility site *j* should be included, given that the Cartesian coordinates of site *i*, (x_i, y_i), and for site *j*, (x_j, y_j) are known. Figure 9 shows the logic structure used to compute core and wedge clique sets membership. This process is based upon the geometrical arrangement of wedges depicted in Figure 8. Function 1, f_1, represents the line separating Wedge 1 & 2 as well as Wedge 4 & 5. Function 2, f_2, represents the line separating Wedge 2 & 3 as well as Wedge 5 & 6. By using the logic structure outlined in Figure 9, members within the r-neighborhood of i can be quickly and easily assigned to a clique set, either the core set or one of the wedge sets.

For example, the first test determines whether site j is a member of the "core" clique constraints or whether it should be assigned to a wedge. If site j belongs to the "core" clique constraints, it is assigned to the core clique constraint and the logic process is terminated. If it is not a member of a core clique, the second logic decision point determines whether it belongs in Wedge Cliques 1 through 3 if the x-coordinate of site j is greater than or equal to the x-coordinate of site i, or 4 through 6 if the x-coordinate of site j is less than the x-coordinate of site i.

If site j is determined to be a potential member of Wedge cliques 1 through 3 a

Figure 9. Logic used to determine members of the Core and Wedge clique sets



further set of logical tests is performed. The first test determines if site j is a member of Wedge Clique 1 by checking to see if site j's y-coordinate is greater than or equal to that of the corresponding y-coordinate of the f_1 line. If it is, it is a member of Wedge Clique 1. If it isn't, then the algorithm checks to see if it is a member of Wedge Clique 2 by determining if the y-coordinate of site j is less than the corresponding y-coordinate of the f_1 line *and* greater than or equal to the corresponding y-coordinate of the f_2 line. If these conditions are met, it is assigned membership to Wedge Clique 2. If it is not, the last logical check point is reached and the y-coordinate of site j is checked to see if it is less than the corresponding y-coordinate of the f_2 line. A similar set of logical tests is conducted to determine whether or not site j is a member of Wedge Cliques 4 through 6 on the other logic branch. Once the logic operations have determined the core and wedge clique members, the following formulation model, the Anti-Covering Location Model – Core and Wedge Clique Constraints (ACLM-CWCC) can be defined.

ACLM-CWCC

$$\text{Maximize } Z = \sum_{j \in S} x_j \tag{16}$$

s.t.

$$x_i + \sum_{j \in CQ_i} x_j \le 1 \quad \forall \quad i \in S \tag{17}$$

$$x_i + \sum_{j \in W_{wi}} x_j \le 1 \quad \forall \quad i \in S \text{ and for } w = 1, 2, 3, 4, 5, \& \ 6 \tag{18}$$

$$x_j \in \{0,1\} \qquad \forall \quad i \in S \tag{19}$$

The ACLM-CWCC is notably different from all previously formulated models in that there is no neighborhood constraint and there is a predetermined limited number of clique constraints. This is because the ACLM-CWCC formulation accounts for all pairwise constraints within *r* distance of facility site *i* in a core or wedge clique. As in previous models the objective (16) and binary integer decision variables restrictions (19) remain the same. Similarly, the core constraint (17) developed by Erkut et al. (1996), which is the same constraint (10) of the ACLM-CCC, has been utilized. Constraints (18) represent the completely new wedge constraints. Additionally, if any core or wedge has no members it does not need to be written out. This further reduces the overall number of constraints required to solve this problem. Formulating the ACLM-CWCC requires at most *n* constraints. The inclusion of the wedge constraints and the elimination of the loose neighborhood constraint can significantly tighten the formulation. Furthermore, conducting the logic tests necessary to formulate the core and wedge constraints is much more computationally tractable than solving node-packing problems to determine maximal clique membership as in Murray and Church (1997a).

In addition to the previously described logic approach, one could include a simple test to determine if a wedge defined for site *i* is a sub-set clique of a core clique for site *j*. If it is, then only the core clique constraint of site *j* need be written and the wedge clique constraint for site *i* can be eliminated as site *i*'s pairwise restrictions will be in a wedge clique of site *j*. In this way the needed number of constraints can be further reduced. This clique set determination is similar to that proposed by Murray and Church (1997a).

There is also an opportunity to reduce the number of needed wedges. This approach involves determining the minimum number of 60 degree wedges that one could use to represent all of the facilities within ½ *r* and *r* of facility *i*. A core clique would only be required if facilities were within ½ *r* of facility *i*. Using this approach could create tight constraints that substantially reduce the total number of constraints necessary to represent the ACLP.

In conclusion, the ACLM-CWCC model utilizes a small number of cliques without the need to solve vertex packing problems, as in Murray and Church (1997a) for Euclidean point datasets. In addition, the ACLM-CWCC contains at most 7*n* tight constraints and does not require a neighborhood constraint as in Erkut et al. (1996). However, the advantage of generating tight wedge and core constraints may not be readily apparent when the number of facilities represented in such constraints is sparse; in that case, the Erkut et al. (1996) formulation (8)-(11) is likely to have the advantage. Results of direct comparison will be presented and discussed in the next chapter. In addition, the geometry relied upon for generating core and wedge cliques has theoretical implication for all of the previously described formulations that use neighborhood constraints.

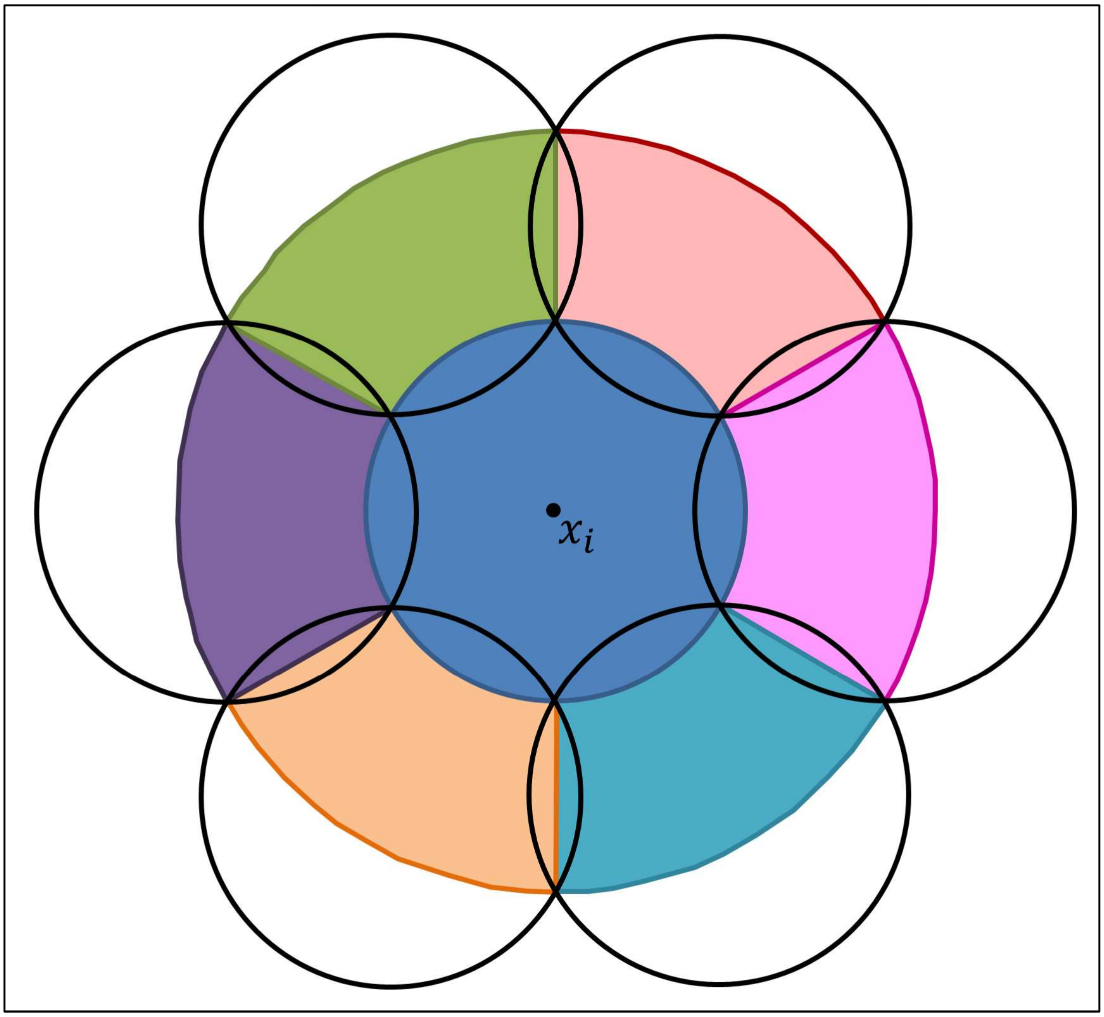
F. *Theoretical implications of Core and Wedge related to Neighborhood Constraints*

In addition to the geometry that informed construction of the ACLM-CWC formulation given above, the geometry has important implications related to the neighborhood constraints of all the previous models *e.g.* constraints of type (2), (9), and (13). Specifically, model ACLM-CWC needs no more than seven clique constraints for each potential site. Each clique constraint represents one of seven zones in the region surrounding a given site *i* (a core and 6 wedges). The seven zones are depicted in Figure 10. The fact that all pairwise conditions can be represented by 7 clique conditions means that if site *i* is not chosen as a site, these seven clique constraints will allow at most one site in each clique constraint to be chosen. Given this observation, one can now prove the following property:

Corollary: a classic neighborhood constraint can be written with an n_i value no greater than 7 without loss of generality for Euclidean based anti-covering location problems.

Proof: given that all pairwise conditions for a given site *i* are represented by 7 clique constraints, and given that these clique conditions will allow at most one site in each clique to be selected, then in total, no more than 7 sites in the neighborhood around site *i* can be selected. This means that the upper limit within the neighborhood is at most 7 and that n_i can be set to a value of 7. QED.

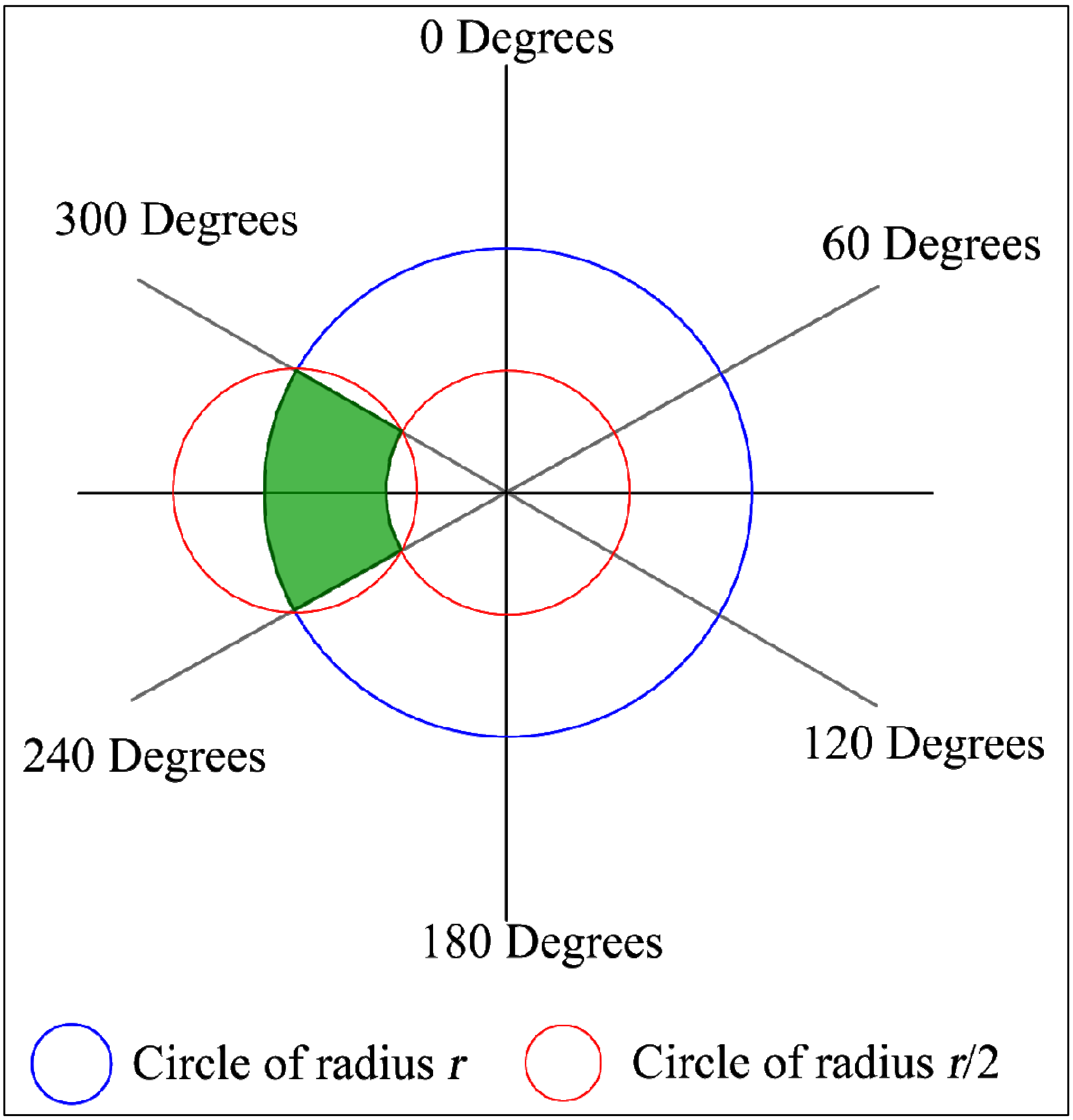
Figure 10. Figure showing the 6 wedges and core.



Therefore, the maximum number of nearest neighbors, n_i, for a neighborhood constraint about site *i* is 7. This represents a valid upper-bound for all such neighborhood constraints. The proof of the above corollary is based upon the notion that there can be at most 7 sites chosen within the neighborhood of a given site. This is because each clique constraint will prevent no more than one site chosen per constraint. But, the fact is there are other restrictions represented in any given problem that will limit the number of selected facilities.

First, one should observe that sites that fall on the circle that is of radius *r* about site *i* are all possible candidates to be selected when site *i* is selected, as each site on that circle meets the separation requirements with site *i*. It is notable to observe that

Figure 11. Regions uniquely within ½ *r* of each other and within *r* of facility site *i*. Example wedge clique is shown in green.



at this distance of exactly *r*, no more than six sites on the circle can be chosen without violating an *r*-distance constraint among them. Thus, at a distance of *r*, it is impossible to have more than 6 neighboring selections when site *i* is selected. This is an alternate way of viewing the proof of the above corollary. The wedge-based clique constraints in themselves, include only sites strictly within the circle of *r* distance of site *i* and fall on or within a given wedge felloe or tire section.

By construction, one can draw a circle of radius *r* about any site *j* in the wedge felloe of *i* and observe that all sites in the core of *i* that are also within the confines of the lines defining that same wedge are strictly within *r* distance of site *j*. This means that if any site *j* in a wedge felloe of site *i* is chosen for a facility, then, restrictions associated with the choice of site *j* will prevent any site chosen within the core set of *i* that falls within the same lines drawn to define the wedge of *i* containing *j*. Given this property, one can now prove the following theorem.

Theorem: a classic neighborhood constraint can be written with an n_i value no greater than 6 without loss of generality for Euclidean based anti-covering location problems.

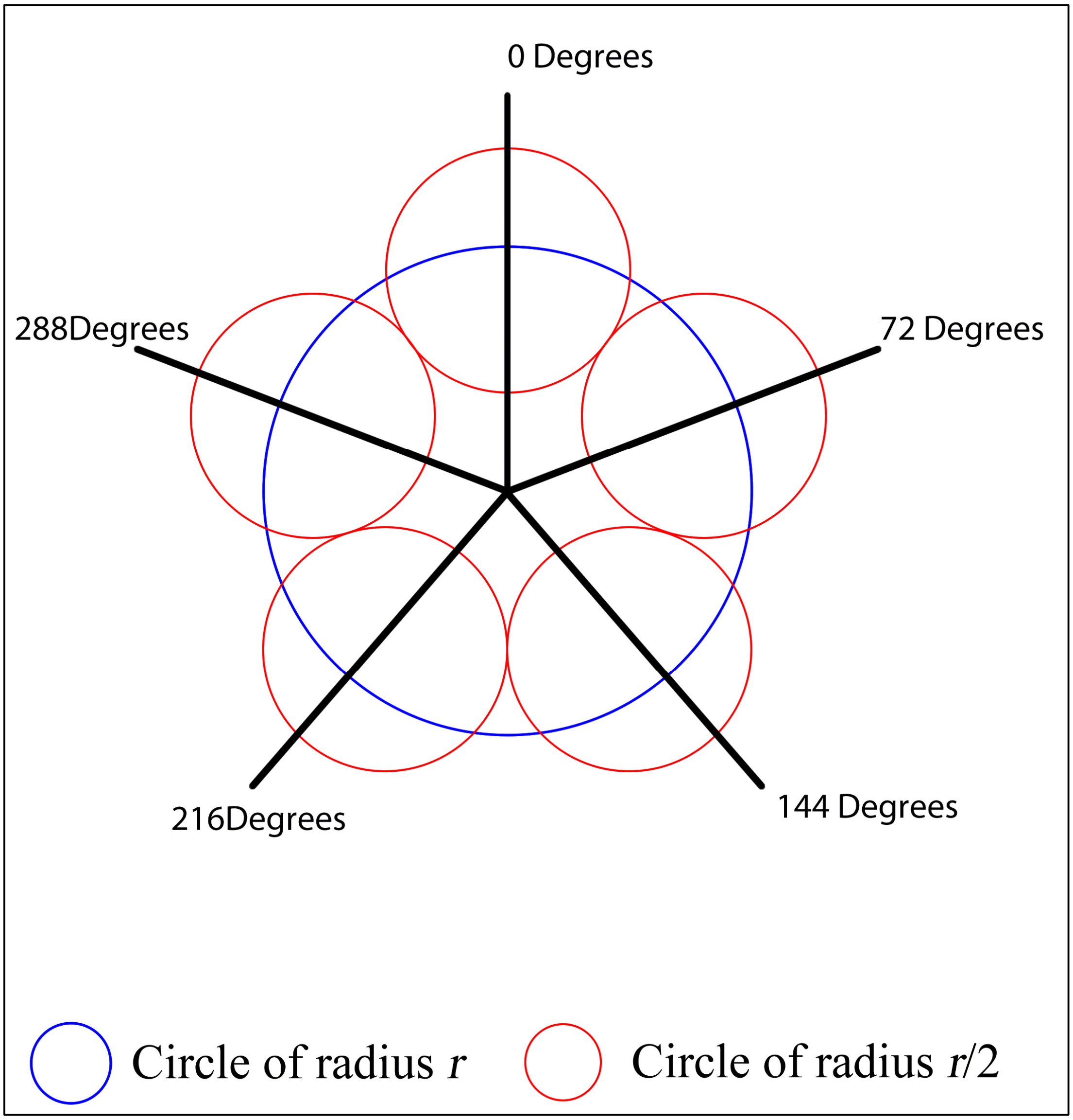
Proof. If selecting a site *j* in a wedge felloe of site *i* prevents the selection of any sites within the portion of the core of *i* defined by the lines that were used to define the wedge, then selecting a site in each of the six wedges will prevent any site being selected in the whole core of site *i*. Thus, no more than six sites within the neighborhood of site *i* can be chosen simultaneously. QED.

However, the above proof applies only to those site locations, *j*, that are *exactly r* distance away from site *i*. One then must ask about the upper-bound for n_i in the case where site locations *j* are less than *r* distance away from site *i*. What might be the maximum number of located neighbors in this case? In this instance, one can show graphically that the maximum number of neighbors that could be simultaneously located within *r* of site location *i* is 5. Figure 12 shows this. If one were to locate these in a symmetric configuration, as in Figure 12, each facility would be located at equal intervals of 72 degrees from one another. Given this geometric configuration, one can now prove the following theorem.

Theorem: a classic neighborhood constraint can be written with an n_i value no greater than 5 without loss of generality for Euclidean based anti-covering location problems.

Proof. If any site *j* that is less than *r* distance away from *i* is selected, it will preclude site *i* from being selected. If site *i* is not selected, then there are opportunities for other *j* neighbors of *i* to be selected and still remain at least *r* distance away from each other. Given the previous proof showing that six sites may be located *r* distance away from *i*, and the fact that a facility may not be located at *i*, it is possible to locate at most 5 sites *within r* of site *i* that may have a facility located such that each of the 5 located facilities are each *r* distance from each other. In this case, one may locate 5 sites greater than or equal to 0.850650808*r* distance of site *i* at equal intervals of 72 degrees about site *i* to generate a symmetrical configuration.

Figure 12. Graphical proof that n_i can be at most 5 with an *r* separation standard



Thus, no more than five sites within the neighborhood of site *i* can be chosen simultaneously. QED.

Therefore the theorem shows that the n_i value of a neighborhood constraint can be limited to no more than 5 when using a model formulation using neighborhood constraints. However, when there are fewer facilities within the neighborhood set, that is $|N_i| < 5$, then n_i should be set equal to $|N_i|$. When generating a model that utilizes neighborhood constraints a simple test, such as $n_i = Minimum(5, |N_i|)$, can be implemented to make the neighborhood constraints as tight as possible.

For example, if one has a neighborhood constraint written as: $90x_1 + x_2 + x_3 + x_4 + ... + x_{90} \le 90$ is mathematically not as tight as a constraint written as: $5x_1 + x_2 + x_3 + x_4 + ... + x_{90} \le 5$. Incorporating this simple property into the previous formulations representing the ACLP which use neighborhood constraints is likely to reduce overall computation time. In fact, the model most likely to experience an improvement when using this particular property is the ACLM-CC formulation first described by Erkut et al. (1996) because it is extremely easy to generate the core clique and determine the remaining neighboring facilities for the reduced neighborhood constraint.

G. Location Set Covering Constraints

In addition to the completely new representation of "Core and Wedge" constraints, another constraint form can be used to further tighten existing formulations for the Anti-Covering Location Problem. This involves the use of a location set covering constraint, initially implemented in the Location Set Covering Problem described by Toregas and ReVelle (1972). The Location Set Covering Problem is a problem where one seeks to identify the minimum number of facilities required to cover a set of demands. Rather than covering a set of demand locations, think of our selection of sites as covering other facility sites. It seems strange at first to think of wanting to cover facility sites in an anti-cover location problem, but in reality it is a property at optimality. That is, at optimality it is impossible to have an unused site that is not within *r* distance of all selected sites. For if such a site existed it could be selected without violating the *r*-separation constraints, and thereby demonstrate that it wasn't optimal. Thus, set covering constraints can be thought of as stipulating that any unselected site must be covered by a selected site, a prospect that must be true at optimality for the ACLP. Incorporating this condition does not restrict an optimal solution from being found, but since set covering constraints are thought to be integer friendly, they may further tighten a problem formulation.

Before the location set covering constraint is presented, some additional notation must be given. In this case one must define the neighborhood of potential facility locations, *j*, that are within *r* of site *i*; that is the neighborhood of sites about facility location *i* or SC_i. More formally:

Additional Notation:

$SC_i = \{ j \in S \mid d_{ij} < r \}$ for all *i* in *S*

Thus, the location set covering constraint is formally defined as follows:

$$\sum_{j \in SC_i} x_j \geq 1 \qquad \forall \quad i \in S \tag{20}$$

Constraints of type (20) are similar to those of the neighborhood constraints e.g. constraints of type (2), (9), and (13).

The neighborhood constraints restrict all of the potential facility locations within the neighborhood of i to be less than or equal to one in value, that is that only one of the facility sites within the neighborhood of i may have a located facility. The location set covering constraint maintains that at least one facility site within r distance of site i must be selected. Though this constraint is structurally redundant, it aids in generating an efficient cut to the polytope of each formulation. Incorporating location set covering constraints into the previous formulations is likely to result in modest to significant performance advances. This constraint is easily added to all of the models and should reduce solution times.

H. Concluding Remarks

This chapter has focused on presenting several constraint structures that can be used to represent the Anti-Covering Location Problem. The chapter described the initial big M constraint formulation presented by Moon and Chaudhry (1984), as well as clique constraint modeling approaches described by both Erkut et al. (1996) and Murray and Church (1996), maximal clique sets (Murray & Church, 1997a), and the completely new Core and Wedge clique constraint model and the Location Set Covering facet.

Erkut et al. (1996) describe a method incorporating Core cliques and the associated neighborhood constraint. Murray and Church (1997a) extended their constraint structuring approach using maximal cliques (Murray & Church, 1996) to generate a maximal clique set. Using maximal cliques requires solving numerous node packing problems, but can reduce the need for numerous pairwise constraints

This reduced constraint represents all sites within *r* distance of each other and within *r* distance of site *i*. The two new constraint structures presented in this chapter, Core and Wedge cliques and the Location Set Covering constraint approaches, have important implications for all formulations currently in the ACLP literature.

The Core and Wedge clique formulation extends the concept of the core clique developed by Erkut et al. (1996) and Murray and Church's (1997a) idea of tight "maximal" cliques. The main difference here is that instead of identifying a singular maximal clique for each potential site location *i*, several small yet dense cliques can be used to represent all pairwise conditions for a given facility site *i* without the need for a loose neighborhood constraint. An added bonus is that no a-priori optimization is required to generate the cliques as in the Murray and Church (1997a) method; only simple geometric tests are required to determine clique membership. Furthermore, the geometric proof supporting the Wedge and Core formulation approach can be further extended to show that the upper bound of the n_i constant in neighborhood constraints can be set at no larger than 5 without loss of generality and can be further reduced if the size of the neighborhood set is less than 5, which will further tighten the neighborhood constraint.

The Location Set-Covering constraint can also be implemented in each of the formulations presented here. The addition of a location set covering constraint is an efficient way to generate a cut facet to the basic matrix of a problem. This cut is likely to reduce solution times. Chapter 5 focuses on the implementation of each formulation presented here.

In conclusion, this chapter has:

- Reviewed previous formulations representing the Anti-Covering Location Problem,
- Presented two completely new constraint representations, and
- Provided an important proof with significant implications related to the neighborhood constraints used in previous formulations.

In the next chapter, two new forms of the Anti-Covering Location Problem will be developed. Computational testing of these two models along with the work presented in this chapter will be presented in chapter five.

IV. New Anti-Covering Models

This chapter outlines two new models related to the anti-covering location problem and discusses the important elements that these models capture that previous formulations do not. As was discussed in the introduction, minimum separation standards have often been used in location modeling. Using a standards based approach, as in the Anti-Covering Location Problem (ACLP), facilities must be kept at least *r* distance from their nearest neighboring facility. To address this problem, all previous anti-covering location modeling approaches have been made with two basic implied assumptions. The first assumption involves the objective of the problem. The second assumption involves the fixed separation distance. Both of these assumptions have affected the way in which dispersion modeling using separation metrics has been conducted and applied.

For example, the first assumption involving the objective function assumes that one wishes to maximize the number of located facilities among a set of potential facility sites or within some bounded region. Several models have been developed to locate the greatest number of facilities separated by some standard, for locating: military defense positions (Chaudhry et al. 1986); estimating the potential impacts of policies on sex offender residence locations (Grubesic & Murray, 2008; Grubesic et al. 2008); designing optimal cut patterns for fabric materials (Wong & Leung, 2009); placing labels on maps (Ribeiro & Lorena, 2008a); designing biological reserves (Williams, 2008); and determining habitat carrying capacity (Downs et al. 2008). The objective of maximizing the number of located objects makes a great deal of sense, particularly for problems involving economic maximization, such as the cut pattern application.

However, many of these problems should also consider the configuration case where the minimum number of facilities is located such that each is separated by a standard. Problems that are likely very sensitive to this nuance, which haven't been considered before, are all of the forms that do not involve an explicit economic objective. For example one should know the minimum number of facilities/objects that are located such as: reserve sites, the minimum population carrying capacity for an area, the minimum number of map labels to place on a map, etc. Thus the first assumption about maximizing the number of objects (facilities) to locate has some implications that have not been previously discussed. This particular issue is addressed in the Disruptive Anti-Covering Location Problem section of this chapter and for which the problem is defined and a location model is presented.

In addition to the first assumption, there is an assumption of a fixed separation standard. If one is locating a series of franchise stores there is a general assumption that the separation standard is constant. However, what if there is a case where one is considering multiple types of facilities that have a varying separation standard? What is the maximum and minimum packing configuration for this case? What if one wishes to relax the separation standard to allow a certain number of "violations", that is, where there are a certain number of facilities that may be closer than r? These issues are addressed in the Modeling Variable Separation Standards section of this chapter.

These two assumptions have un-intended or previously un-accounted for modeling consequences that have not been addressed in the anti-covering location problem literature to date. The rest of this chapter is devoted to addressing these modeling shortcomings to improve dispersion modeling within the context of separation standards. Each section will provide a description of the problem and formulations representing the problem.

A. The Disruptive Anti-Covering Location Problem

The anti-covering (or *r*-separation) location problem (ACLP) involves maximizing the set of located sites, such that no two located sites are closer than a specified distance, time, or other standard of each other. This problem can be defined on a bounded continuous region or a discrete set of sites. When defined on a bounded continuous domain it is generally assumed that all facilities must be located within the region and be further than *r*-distance from the boundary and *r*-distance from each other. The solution to this problem is sometimes referred to as a packed arrangement.

There may be many configurations to a problem instance in which all facilities are at least the prescribed *r*-distance apart from each other. Those arrangements which involve the maximum number of located facilities are optimal ACLP solutions. Those solutions that use fewer than the maximum possible number of located facilities fall into two cases: 1) sites exist where it is possible to locate additional facilities and still maintain the *r*-separation constraints; and, 2) all remaining unused sites are too close to an existing facility or boundary so that no further sites can be added to the solution without violating the *r*-separation constraints. This chapter section deals with this second type of solution.

If one is considering the case where all facilities must be separated by at least r such that no other facility may be packed in, a logical question to ask is the following: “What is the smallest number of facilities needed and their placement such that no remaining sites can be used without violating one or more *r*-separation constraints?” The basic element to this problem is to find the smallest configuration that blocks to the greatest extent possible a maximal packing. This problem case can be described as the Disruptive Anti-Covering Location Problem (DACLP).

The formulated model to address this problem must identify solutions that prohibit a maximally packed configuration; such solutions can be thought of as being disruptive to a maximally packed solution. The importance of this problem is both theoretical and practical. From either perspective, optimal solutions to the DACLP define a lower bound on the number of facilities that can be placed without violating the *r*-separation constraints as well as pre-empt any additional facilities from being feasibly added. This is an important consideration, particularly in problems where a lower-bound packing arrangement should be considered.

Most of the applications of the anti-covering location problem (ACLP) entail the use of one of the models described in chapter 3. Prospective sites are identified in advance as “discrete” locations, representing centers of raster cells (Church, 2013), commercial parcels (Grubesic et al. 2012), or nodes of a network. Murray and Church (1997a) have shown that the discrete anti-cover problem is an equivalent problem to the vertex packing problem on a network or the maximal independent set problem on a graph, and therefore is NP hard.

Optimal solutions to the anti-covering problem represent the largest number of facilities that can be simultaneously located while keeping each of them at least a minimum distance, *r*, from each other. Unfortunately, there can be circumstances in which a maximum packing is disrupted; that is, not optimally packed. They may be disrupted by earlier residential choices, already established crane nests and territories, or by poor choices in already located franchisee establishments. Whether maximal packing arrangements are disrupted by accident, happenstance or by intent, such disruption and the potential impact of disruption should be of interest when using this type of model. Additionally, given that many problems have used the ACLP to find maximal packing configurations, and Wei and Murray (2012) have shown that spatial uncertainty plays a role in determining various packed configurations, one should also consider the lower bound or maximally disruptive case as well within the context of spatial uncertainty.

Understanding the configuration and number of sites that can be located is particularly useful for applications related to habitat nest/den site modeling, modeling feasible residence locations for sex-offenders, modeling franchise store location, or any other application for which the ACLP has been used. In the next section a basic way in which to represent the Disruptive Anti-Covering Location Problem (DACLP), formulated as an integer programming problem called the Disruptive Anti-Covering Location Model (DACLM), is described. Following this, a brief discussion on packing solutions using the DACLM and ACLM and some other important modeling details related to the DACLM.

1. Formulating a model for the disruptive anti-covering location problem (DACLP)

A feasible solution to an anti-covering location problem must have all facilities placed at least *r*-distance apart. If a feasible solution to an anti-covering location problem also has the property that no additional sites can be chosen without violating one or more separation constraints, then one calls that solution a *proper* solution. An optimal solution to an ACLP is a *proper* solution which involves locating the largest number of facilities possible. The disruptive anti-covering location problem has the opposite goal as that of the anti-covering location model. It can be formally defined as:

What is the minimum number of facilities and their arrangement such that each facility is separated by at least r-distance from all other facilities and no remaining sites exist in which another facility can be added without violating one or more of the separation conditions.

Thus, it involves finding a *proper solution* which involves the location of the smallest possible number of facilities. If one defines:

$p_{\max} =$ the number of facilities deployed in an optimal anti-covering solution, and

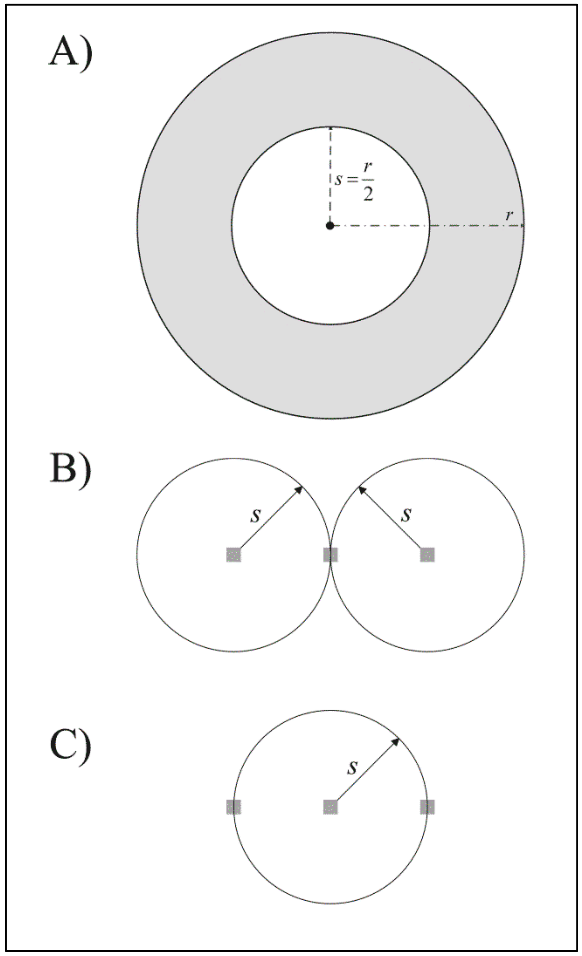
$p_{\min} =$ the number of facilities deployed in an optimal disruptive anti-covering solution,

then all proper solutions to a given problem instance will deploy a number of facilities that can be bounded as follows:

$$P_{\min} \leq \text{number of facilities used in a proper solution} \leq P_{\max} \tag{21}$$

Figure 13 contains three parts showing: the minimum separation distance *r* and how packing circles of radius *s* are related, a proper ACLP solution considering three potential facility locations, and a proper DACLP solution considering the same three potential facility locations. Figure 13A shows how the minimum separation standard *r* is related to packing circles of radius *s*, where *s* = *r*/2. Figure 13B depicts the case where there are three site locations represented as small squares. If the left most site and the right most site are selected for facilities then one can see that their disks of radius *s* touch, but do not overlap. Thus, this solution is feasible. Further, the middle site is too close to the other two sites as its disk would overlap with the others. Thus, this solution is an optimal ACLP solution. Figure 13C depicts a different solution where the middle site has been chosen for a facility. The choice of this middle site would preclude the choice of any additional site for a facility because the remaining two sites are too close. Thus, this solution is also proper. This solution is an optimal disruptive anti-covering solution.

Figure 13. A) Example of r separation standard and packing circle of radius s. B) A proper optimal solution to the ACLP. C) A proper optimal solution to the DACLP for the same sites in B.



The objective of this section is to define a model which can be used to solve the disruptive anti-cover problem and thereby calculate $P_{\min}$. Using the notation that has already been introduced, one can formulate the disruptive anti-cover model following the form used by Moon and Chaudhry (1984) as follows:

DACLM

$$\text{Minimize } Z = \sum_{j \in S} x_j \tag{22}$$

s.t.

$$n_i x_i + \sum_{j \in Q_i} x_j \le n_i \qquad \text{for all } i \in S \tag{23}$$

$$x_i + \sum_{j \in Q_i} x_j \ge 1 \qquad \text{for all } i \in S \tag{24}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in S \tag{25}$$

The objective (22) involves minimizing the number of sites selected for facility placement. Constraints (23) ensure that each located facility is separated by at least *r*-distance from all other located facilities. Constraints (24) basically require that the resulting solution is a feasible proper packing solution. Constraints (24) require that each unused or unselected site is less than *r* distance away from a located facility. In essence, constraints (24) force the model to locate enough facilities that each unused site is close enough to a located facility that its choice as a facility site would violate a separation standard. This means that constraints (24) establish that the solution must be proper; that is no feasible site exists within the located configuration that is *r*-distance or further from all other selected sites.

One should recognize that constraint (24) is the Location Set Covering constraint first described by Toregas and ReVelle (1972) which was previously discussed in Chapter 3; see constraints (20). Constraints (24) simply ensure that at least one site is chosen nearby to site i (strictly within r distance) or site i itself is chosen for a facility. Constraints (25) are the binary integer restrictions for the facility site location variables. Altogether, the model involves finding a feasible, proper anti-covering solution that uses the smallest number of facilities.

The above model can also be formulated with pairwise (5) and higher ordered clique constraints (7) instead of constraints (23), as discussed in Chapter 4. This includes any of the other forms discussed in Chapter 4 such as: core cliques, maximal cliques, and core and wedge cliques. In many circumstances a hybrid model using both types of constraints may prove to be the best when using off-the-shelf commercial solvers. The model as formulated above is an integer programming problem. Because it is a combined form of the vertex packing problem and the set covering problem, it is functionally related to the class of NP-hard problems. The fact that it is related to two complex problems virtually ensures that the above model will not always be solvable to provable optimality. As the number of sites increases, it appears that the difficulty of the problem will tend to increase, and for large problems, one may have to resort to heuristic approaches. This will be discussed in Chapter 5. In the following sub-section, the subject focuses on the solution of disruptive anti-cover location problems using the model described above. In the next sub-section, details associated with the application of the ACLP and DACLP models applied to two different data sets are presented.

2. A comparison of DACLP and ACLP solutions

In this section example of solutions associated with solving the DACLP are provided. The presentation here is not meant to be exhaustive, but more illustrative of what can be learned from solving the DACLP as compared to the ACLP. Both ACLP and DACLP problems were solved using two different spatial problems over a range of separation distances. The first dataset is the well-known Swain (1971) data set of 55 nodes. Each node represents a potential facility location for a postal delivery zone. The second set is a 372 node dataset from Ruggles and Church (1996). This dataset contains several known Aztec cities, villages, and hamlets and other settlement locations that are believed to have existed prior to the arrival of the Spanish conquistador Hernán Cortés in 1519. The Aztec dataset in particular is particularly interesting because of the potential implications related to central place theory with regard to packing, service access, and regional centers. This research in particular could support the previous work related to optimal dispersion and central places conducted by Curtin and Church (2007).

The ACLP model was formulated and solved using objective form (1) and constraints of type (3) and (9). The DACLP was formulated and solved using objective form (22) and constraint types (9), (24), and (25). These two formulations involve a “neighborhood” or nodal style of separation constraints (i.e. (2) and (9)). The approach of Yoshimoto and Broadie (1994) is used to define the main coefficient for such constraints in this case. No attempt to test other formulations, especially a hybrid form involving clique-based separation constraints, was conducted here as this was not the central theme of this chapter. The Xpress modeling language was used to set up each problem and then solved using the Mosel 64-bit solver version 3.4.3. Xpress and Mosel are products of FICO, or the Fair Isaac COrporation. A 2.4 GHz Intel Xeon workstation with 12 gigabytes of memory running the Windows 7 operating system was used to solve the DACLP and ACLP using the two spatial datasets.

Table 1 presents the results obtained when solving a selected set of *r* values of anti-covering and disruptive anti-covering problems applied to the 55 node data set of Swain (1971). Solution times are not included as all but one problem were solved in less than the smallest time increment of the solver and were reported by the solver as 0.00 seconds. The ACLP problem with $r = 7$ required 0.01 seconds. Sites that were closer than *r*-.00001 distance units were considered to be too close and were prevented from being simultaneously used in a solution. Solutions were obtained by setting specific *r*-separation distances, ranging from a low of 4.0 to a high of 60.0. For each specific separation distance, the number of facilities located by the anti-cover model and the disruptive anti-cover model are listed.

For example, for a separation distance of 10.0 the anti-cover model packed 17 facilities across the 55 sites and kept all facilities separated by at least 10 distance units. For that same distance, the disruptive anti-cover solution involved placing 9 facilities. That is, it is possible to locate 9 facilities in such a manner as to keep all of the facilities separated by at least 10 distance units where all other sites are too close to chosen sites to allow additional sites to be selected. The level of disruption for this case is quite substantial, a 47% reduction. It is important to observe that the difference between what can be located in the packing case (anti-cover) vs. the disruptive case (disruptive anti-cover) is quite small for relatively small distances as well as for relatively large distances. In the mid-range of distance values, there is a considerable difference between what each model is able to locate, a difference that ranges from 30% to a high of 75%. This is a substantial difference and is of critical importance.

Table 1. Results associated with solving the ACLP and DACLP on the Swain Data set. Note that the ":" indicates an incremented number of facilities; *e.g.* 1:4 is equivalent to 1, 2, 3, 4.

Separation Distance, *r*	ACLP Obj	Sites selected by ACLP	DACLP Obj	Sites selected by DACLP	% Difference Between ACLP & DACLP
4.0	45	3, 5:7, 10, 12, 14:28, 30, 32, 33, 35:55	41	7:12, 14:28, 30, 35:41, 43:46, 48:55	8.89%
5.0	38	5, 8, 10, 12, 14:19, 21:23, 26:28, 30, 35:55	30	5, 8, 10, 12, 14:18, 20, 24:29, 32, 33, 36, 37, 39, 40, 42, 43, 49:54	21.05%
6.0	33	4, 8, 10, 12, 14:19, 21, 26:28, 33, 35:41, 44:52, 54, 55	23	1, 3, 6, 12, 14, 16, 17, 20, 24, 26:29, 32, 36, 37, 39, 40, 46, 51:54	30.30%
7.0	25	4, 10, 12, 14, 15, 21:23, 25:28, 31, 34, 35, 37:40, 46, 47, 49:52	17	8, 9, 14, 16, 17, 20, 24:26, 31, 37, 40, 43, 51:54	32.00%

8.0	21	6, 8, 12, 14, 20, 23, 25:27, 31, 33, 35, 37:40, 46, 47, 50:52	13	4, 14, 18, 21, 25, 28, 33, 35, 40, 43, 50, 52, 54	38.10%
9.0	18	3, 6, 12:14, 18, 20, 23, 24, 27, 33, 37:40, 48, 51, 52	11	3, 9, 14, 21, 26, 28, 33, 48, 50, 54, 55	38.89%
10.0	17	1, 10, 12, 14:16, 23, 26, 30, 35, 39, 40, 43, 49, 51:53	9	4, 17, 20, 27, 30, 36, 52, 53, 55	47.06%
11.0	14	2, 10, 12, 18, 24, 27, 33, 39, 40, 43, 48, 51:53	7	5, 16, 20, 23, 36, 52, 55	50.00%
12.0	13	4, 14, 18, 24, 25, 28, 33, 40, 43, 51:54	7	5, 16, 20, 23, 26, 52, 55	46.15%
13.0	11	13, 14, 24, 28, 33, 37, 40, 41, 51, 52, 54	6	8, 20, 22, 36, 39, 52	45.45%
14.0	11	14, 15, 24, 28, 33, 37, 40, 44, 49, 52, 54	5	18, 22, 49, 53, 55	54.55%
15.0	10	14, 26, 28, 33, 37, 39, 42, 46, 49, 52	5	18, 20, 22, 43, 53	50.00%
20.0	7	8, 27, 28, 35, 40, 50, 51	3	12, 41, 46	57.14%
25.0	5	28, 35, 39, 51, 52	2	4, 14	60.00%
30.0	4	14, 26, 51, 52	1	1	75.00%
40.0	3	14, 35, 52	1	18	66.67%
50.0	2	14, 51	1	28	50.00%
60.0	1	28	1	28	0.00%

Figure 14 and Figure 15 depict opposing solutions, one for the ACLP and one for the DACLP when using the separation distance of 12.0. Both figures display all 55 potential facility sites as black dots. Selected facility sites are shown as red squares with the separation distance of radius *r*, in this case 12.0, represented as a black circle with gray fill drawn around each selected facility site. Note that in each figure all unused sites are within the minimum separation distance of 12.0 from one or more located facilities. From Table 1, it can be observed for the separation distance of 12.0 that the ACLP solution involves the location of 13 facilities. Figure 14 depicts this ACLP solution. In Figure 14 observe that each selected site is outside all circles except for the one representing that site. This means that the pattern meets all separation requirements. Figure 15 presents the related disruptive solution that involves the placement of only 7 facilities, which is 6 facilities fewer, a 46.15% reduction, than what could be located in the optimal anti-cover solution. These two solutions capture the range in which proper solutions exist for the separation distance of 12.0.

Figure 14. An optimal anti-cover solution associated with the separation distance of 12.0 involving the location of 13 facilities.

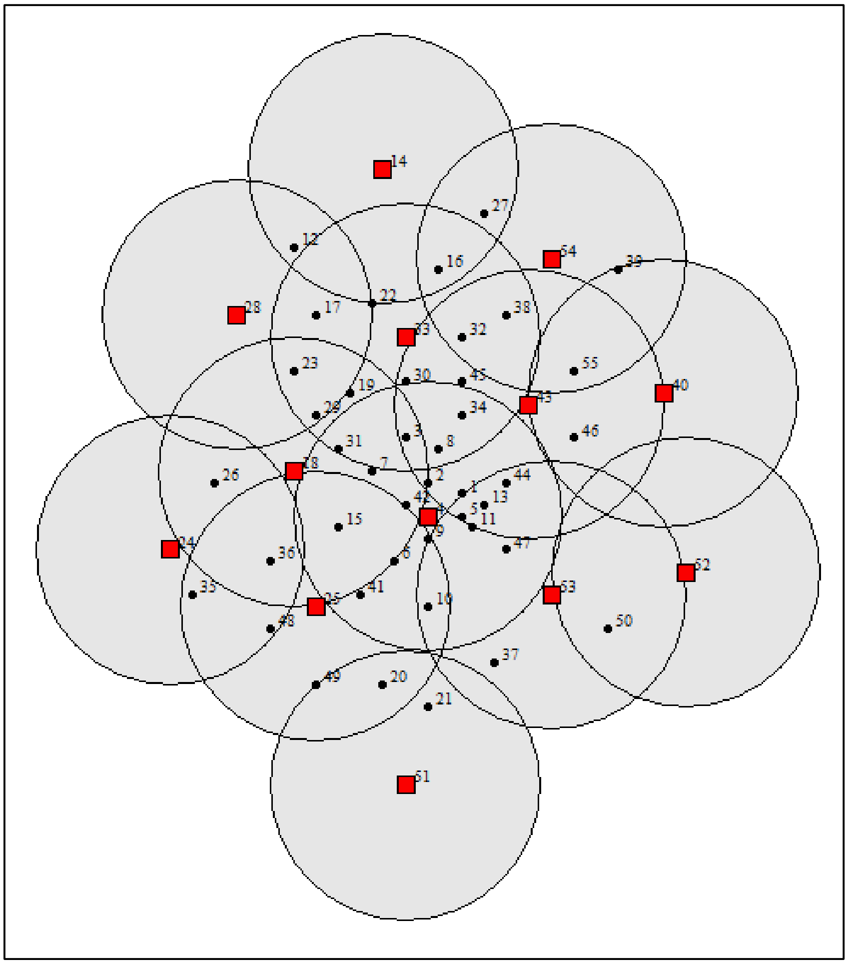
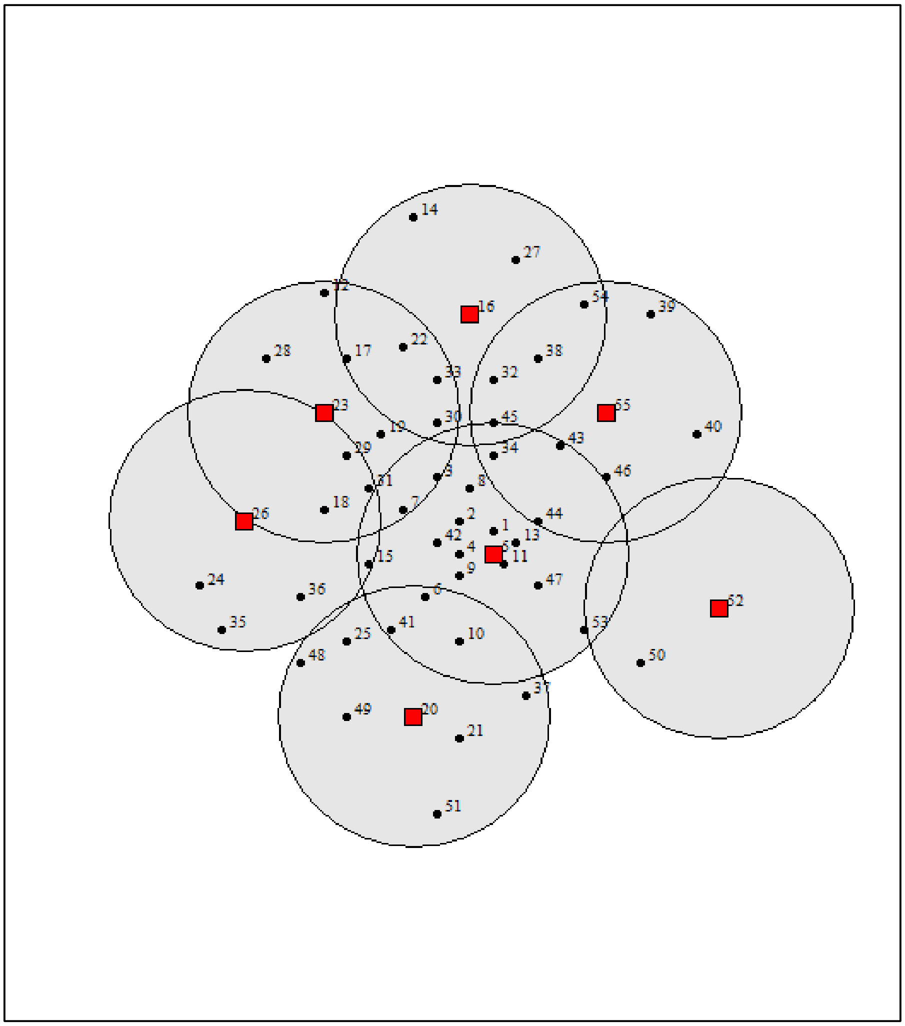


Figure 15. An optimal disruptive anti-cover solution associated with the separation distance of 12.0 involving the location of 7 facilities.



In addition to the solving the ACLP and DACLP models on the 55 node dataset (Swain, 1971), the models were also applied to the larger dataset of 372 nodes (Ruggles & Church, 1996). Table 2 presents the results of these problems using separation distances that ranged from 2.0 to 15.0. For the distance of 2.0, the anti-cover model involved locating 110 facilities, or selecting more than 1 out of 4 sites for a facility on the average. For that same distance, the disruptive model was able to find a proper solution which located only 68 facilities, a 38.18% reduction, or less than 2 out of 10 sites for a facility on average. Over the range of distance values, the disruptive case often differs considerably from the packed case in terms of the number of facilities that were located ranging from 38.18% to 63.64%. Though the focus of this chapter is not on solution times, solution times for both problem types are given in the table as well. This is included to show that the computational effort required to solve the DACLP is much less. This is exemplified using this larger data set; it can be seen that the disruptive case can be solved in considerably less time than the original anti-cover problem.

In most cases the disruptive model was solved in less than a tenth of the time needed for the packing (ACLP) model. Although it may be possible to reduce computational times below what is reported here by using a selected set of clique constraints or a hybrid of cliques and neighborhood constraints, it is likely that the disruptive model is easier to solve in general as compared to the classic anti-covering problem. It should also be mentioned that modeling languages also exact a cost in terms of set-up and execution time when using maximal cliques or a hybrid approach; however, this cost is often outweighed by the value in increased computational efficiency. Greater discussion related to why the DACLP solves faster than the ACLP and a comparison of solution approaches using modeling languages and alternative methods is presented in greater detail in Chapter 5.

Table 2: Results associated with solving the ACLP and DACLP on the 372 Node dataset of Ruggles and Church (1996)

Separation Distance, *r*	ACLP Objective	ACLP Solution Time in seconds	DACLP Objective	DACLP Solution Time in seconds	% Difference Between ACLP & DACLP
2.0	110	0.15	68	0.00	38.18%
3.0	68	0.25	35	0.15	48.53%
4.0	45	0.80	22	0.10	51.11%
5.0	32	2.50	16	0.20	50.00%
6.0	26	1.30	11	0.20	57.69%
7.0	21	1.90	9	0.20	57.14%
8.0	17	3.20	7	0.20	58.82%
9.0	14	2.40	6	0.20	57.14%
10.0	11	2.40	4	0.30	63.64%
11.0	10	2.40	4	0.30	60.00%
12.0	9	3.10	4	0.30	55.56%
13.0	8	4.10	4	0.30	50.00%
14.0	8	5.10	3	0.40	62.50%
15.0	6	6.80	3	0.40	50.00%

3. Searching for stable levels of possible disruption

A proper solution to the anti-cover problem maintains a minimum separation of *r* distance between any pair of facilities and where no additional facilities may be located without violating a separation constraint. As described in the previous section, the range of proper solutions will involve a number of facilities which fall within the range:

$$P_{\min} \leq \text{number of facilities used in a proper solution} \leq P_{\max}$$

Where $P_{\max}$ and $P_{\min}$ can be generated by solving the ACLP and DACLP respectively. What is not known is whether solutions to a specific problem exist within the range or whether solutions exist only at the upper and lower bounds. In addition, it is unknown from the outset for a given problem whether multiple proper patterns exist at the bounds or within the range. One can define for any value of *p* within the range for which one or more proper solutions exist, a stable level. It makes sense to identify where stable levels exist between the upper and lower bounds, $P_{\max}$ and $P_{\min}$. One approach to identifying whether a stable level exists is by appending the following constraint to the model described by conditions (22)-(26):

$$\sum_{j} x_j \geq p \qquad \forall\ j \in S \tag{26}$$

Constraint (26) maintains that at least *p* facilities are to be deployed. The value of *p* can range from $P_{\min}$ to $P_{\max}$. If an optimal solution to this problem deploys exactly *p* facilities, then *p* represents a stable level of disruption. If not, then the solution deploys some number of facilities $p^* > p$. This means that all values of *p* strictly between p^* and *p*, including *p* itself, are not stable levels for the problem instance. Thus, in solving for the existence of stable levels, it makes sense to first solve with the bound of $p = p_{\min} + 1$. After solving that problem, each subsequent problem is defined by setting $p = p^* + 1$ until $p^* = p_{\max}$ or $p^* = p_{\max} - 1$. This strategy can be used to efficiently solve for the stable levels of *p*. This strategy is used to solve several example problems that were presented in the previous section. Essentially a constraint was added to the problem and solved in sequence, based upon the value of *p* specified in constraint (26) for the previous iteration.

Table 3: Results associated with solving for stable levels of the DACLP on the Swain data set.

Separation Distance, *r*	DACLP Objective	Stable values of *p* found between strict ACLP and strict DACLP	ACLP Objective	Percent Difference Between ACLP & DACLP
5.0	30	31,32,33,34,35,36,37	38	21.05%
10.0	9	10,11,12,13,14,15,16,17	17	47.06%
15.0	5	6,7,8,9	10	50.00%
20.0	3	4,5,6	7	57.14%
25.0	2	3,4	5	60.00%
30.0	1	2,3	4	75.00%

Table 3 presents the results for the search for stable levels for disruptive anti-covering when solving for selected separation distances on the Swain data set. Several problems were solved for stable disruption levels for 6 different separation distances ranging from 5.0 to 30.0 with increments of 5.0. Each of these problems solved in less than 0.00 seconds. Altogether, stable disruptive solutions for all 26 possible cases were identified. At first this seems to be somewhat counterintuitive, however, after the fact it seems entirely reasonable, as one should be able to make just the needed amount of adjustment to a disruptive pattern so that exactly one more facility can be added, bringing it to the next stable level. Of course this will not hold for all problems, but it is likely that this would be commonplace rather than the exception.

4. A note on aiding disruption

Together, the DACLP and ACLP models can be used to generate the range of proper solutions to a given problem instance. A constrained version of the DACLP model can then be used to generate stable levels within the range of feasible, proper values of p. As stated before, disruption can be accidental, natural, or intentional. For example, the ACLP can be used to generate an arrangement that maximizes the number of Sandhill cranes that can be supported in a bounded area of suitable habitat, where crane nests are separated by a minimum distance of r (Downs et al., 2008). But current nest patterns may not be optimal and together they may thwart the existence of a larger number of nests being supported. The same can be said for a problem of locating liquor stores (Grubesic et al., 2012). If liquor stores are to be located at least r distance apart from each other and at least a certain distance from special areas, like schools, then an existing store pattern may "disrupt" the location of new entrants. It also may be possible for a new entrant to locate in such a manner as to prevent others from locating nearby and effectively increase their neighborhood market size (Church and Bell, 1990). Such a circumstance leads to two types of location questions: 1) what is the best location that a new entrant can make within the separation constraints and effectively develop the largest "hegemony" against others in possible encroachment? and; 2) how many facilities are needed and what are their locations that will aid resulting disruption the most?

Both problems are of considerable interest. The first of which can be solved by a site search, looking for the site which effectively controls the greatest amount of surrounding area. The second problem is of considerably greater complexity and can be formulated as a bi-level integer programming problem. The model representing this bi-level structure can be formulated as follows:

$$\text{Minimize} \quad G \tag{27}$$

s.t.

$$\sum_j y_j = p_l \tag{28}$$

$$M(1 - y_i) \geq \sum_{j \in Q_i} y_j \qquad \text{for all } i \in N \tag{29}$$

$$y_j \in \{0,1\} \qquad \text{for all } j \in N \tag{30}$$

$$G = \text{Maximize} \sum_j x_j \tag{31}$$

s.t.

$$M(1 - x_i - y_j) \geq \sum_{j \in Q_i} x_j \qquad \text{for all } i \in N \tag{32}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in N \tag{33}$$

where the additional notation is as defined as follows:

$G =$ the maximal number of facilities that can be placed after the selection of p_l preemptive sites

$p_l =$ the number of facilities being placed in which to maximally disrupt an anti-cover solution

$$y_j = \begin{cases} 1, \text{if site } j \text{ is selected for as a preemptive disruptive site} \\ 0, \text{ if not} \end{cases}$$

The above bi-level optimization model involves a leader and a follower. The leader makes the decision to locate p_l facilities as maximally disruptive sites. The portion of the model that is the leader consists of constraints (28) through (30). These constraints should be recognized as being similar to those implemented in the formulation representing the disruptive anti-cover location problem (DACLP). The follower solves for the optimal anti-cover solution, given that p_l facilities have already been placed. The follower portion consists of constraints (32) and (33); these constraints are similar to those defined in the formulation representing the ACLP. A description of the technical workings of the constraints of this leader and follower bi-level optimization model is as follows.

Whatever the leader selects, the follower responds with the best anti-cover pattern that can occur given what the leader has selected. The leader minimizes the resulting level of facility placement *G*, formulated as (27) & (31), while making the location decisions for p_l facilities (28). The siting decisions, y_i, must be binary in value, constraints (30), and they must satisfy the separation requirements (29). The value of *G* is defined by a modified anti-covering model. Constraints (32) ensure that selected sites by the leader and the follower must be at least *r*-distance apart from each other. The fact that the right hand side will always be greater than or equal to zero will ensure that the same site will not be selected by both the leader and the follower (*i.e.* it is impossible for a given site i for $x_i = 1$ and $y_i = 1$ at the same time). Finally, constraints (33) restrict the site selections to be binary for the follower.

The solution to this model (27)-(33) will identify a set of p_l sites which disrupt the completed anti-covering solution the most. However, solving bi-level models can be a complex task that often involves significant computational effort, even for small to moderately sized problems (Scaparra and Church, 2008). Thus, solutions to this formulation are not discussed in Chapter 5. Nevertheless, the model is formulated here as an important branch that future researchers should explore.

B. Other forms of r-Separation

All prior formulations of the ACLP have focused on maximizing the packed configuration such that all facilities are separated by at least some standard *r*. Only the paper by Murray and Church (1997b) even considered the issue of site benefit in a formulation they provided. Yet, they did not determine if any change in site benefit had an impact on the overall packing configuration, as that was not the objective of their paper. However, what if the quality of a site requires a separation standard that is smaller or larger than another site? What if a particular site or configuration allows for a certain number of violations of the separation standard? These are two questions that have not been fully addressed in the literature and are of concern when modeling phenomena. The rest of this section defines several ways that a minimum separation standard: can be relaxed as a percentage of sites located near a site; or where the separation standard varies by site benefit. This section describes a few additional formulations that either relax the packing restrictions or consider separation standards as a function of site benefit.

1. The generalized or "almost" *r*-Separation ACLP

The Generalized or Almost *r*-Sep Problem is a problem in which the distance constraints are relaxed for a percentage of potential sites within radius *r*. In other words, some locations will be allowed to violate the explicit *r* separation criterion. This is an interesting problem in that it represents reality in that territories of nearby territorial species (*e.g.* fox, California Spotted owl, and fishers) often overlap modestly. However, one again must ask which locations one would allow to violate the *r*-Sep rule; are they random locations across a landscape? Are they in an area that has a high density of habitat/customer support? Or, is there a strategic decision that has been made to allow for incursions into territory; say for a family group that allows siblings to locate near them or for a retail chain that locates two of their stores close together, such as Target, or coffee houses such as Starbucks? These are all questions that would need to be explored to provide a definitive answer. I believe, however, that a general model such as the one presented below could be effective in comparing outcomes based upon: random dispersion, additional landscape support, or strategic influences.

If one allows for some facilities to be a bit closer than the desired r-separation constraint, then it may be possible to select and pack more facilities into a given region. It may also allow for one to locate at sites which are preferred over others but would not ordinarily be chosen because they are a bit too close. In addition, one should limit such "incursion" as separation standards are often suggested for solid reasons. Consider then the following problem definition:

Generalized r-*Separation problem*: Maximize the weighted benefit of sites selected for a configuration of facilities, where in general each site must be at least αr distance from each other and where for each sited facility, at most one of their neighboring facilities can be closer than r distance.

This problem is called the generalized r-separation or the generalized ACLP. The value of α is to be some value less than or equal to 1. The above generalized problem represents the classic ACLP when α is equal to 1 and all site values are equivalent. This new problem definition allows for the fact that only one neighboring facility of a given located facility can be somewhat closer than the standard r separation distance. As the value of α would probably be on the order of say 0.90, even the closest neighboring facility to a given facility will be forced to be close to the standard r-separation distance. What this problem allows for is a modest flexibility in applying the separation standard as well as encourages the selection of those sites which are weighted more than others.

The following formulation uses the notation that has been previously defined as well as:

b_i is the benefit of locating at location *i*.

The Generalized *r*-Separation or "almost" *r*-Separation model can be thus formulated as follows:

$$\text{Maximize} \sum_{j} b_j x_j \tag{34}$$

s.t.

$$(n-1)x_j + \sum_{k \in N_j} x_k \leq n, \text{ where } N_j = \{j \mid \text{site } j \text{ is within } r \text{ of } k\} \tag{35}$$

$$nx_j + \sum_{k \in N_j} x_k \leq n, \text{ where } N_j = \{j \mid \text{site } j \text{ is within } \alpha r \text{ of } k\} \tag{36}$$

$$\sum_{j \in N_j} x_j \geq 1 \quad \text{for each } k, \text{ where } N_j = \{k \mid \text{site } k \text{ is within } r \text{ of } j\} \tag{37}$$

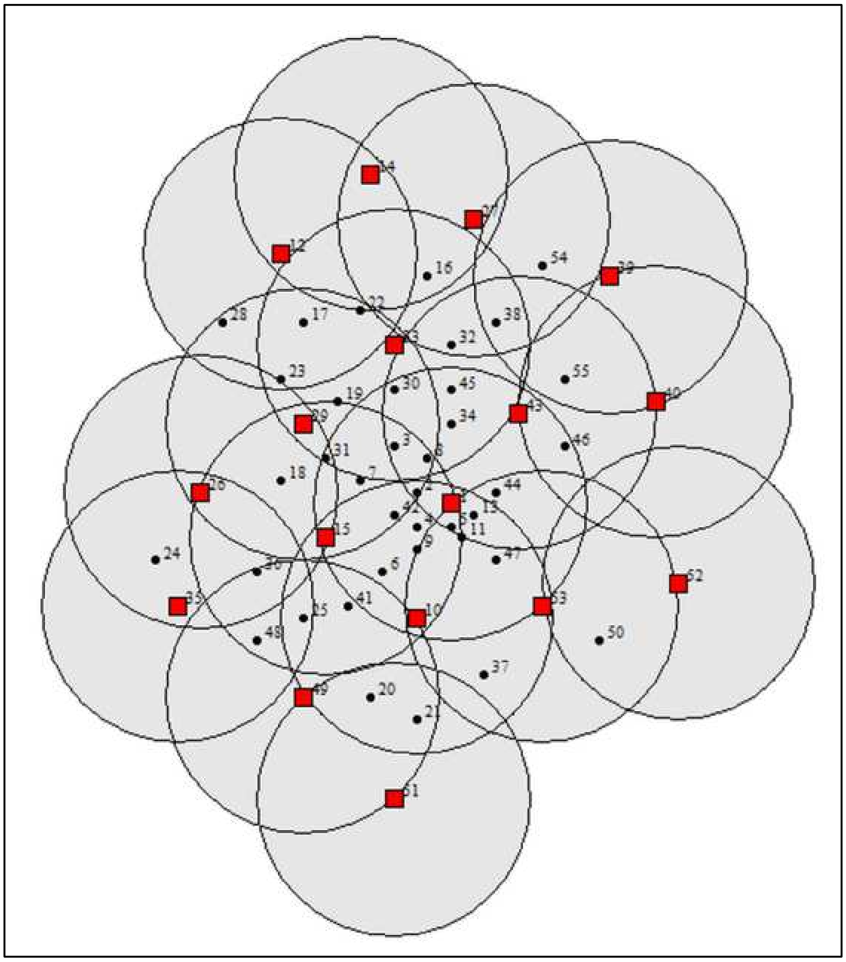
$$x_j \in \{0,1\} \quad \text{for each} \quad j \tag{38}$$

The objective (34) is to maximize the number of packed facilities with the greatest benefit. Constraints of type (35) enforce the separation of facilities. In this case at most one facility may be allowed to locate closer than *r* distance of a located facility. Constraints (36) work in conjunction with constraints (35); this constraint limits any selected facility to be closer than α of *r* distance to any other located facility. Constraints of type (37) require that any facility *i* within the r neighborhood of facility site *j* must have at least one facility located within its neighborhood. Though these constraints are technically redundant, as mentioned in Chapter 3, they aid in generating a tighter relaxed linear programming solution. Constraints (38) restrict the location site decision variables to be binary integers.

This model is likely to be helpful particularly for habitat modeling and franchise store modeling where good habitat or market areas provide enough support for the den/franchise store. Figure 16 shows an optimal solution obtained using the above formulation with a separation distance, r, equal to 12, α equal to 0.8 and where all benefit values were set equal to one. For this case, 17 facilities have been located at sites: 1, 10, 12, 14, 15, 26, 27, 29, 33, 35, 39, 40, 43, 49, 51, 52, and 53. The solution differs from that of Figure 14, where 13 facilities have been located. The optimal solution to the Generalized ACLP has 7 sites in common with the optimal solution to the ACLP; sites 14, 33, 40, 43, 51, 52, and 53. However, allowing for a 20% relaxation of separation distance for only the closest neighboring facility, four additional facilities can be placed using the almost r-Separation formulation. This is apparent in Figure 14 as there are subtle shifts in where a facility is located in the Generalized ACLP vs. the ACLP. In addition, one can observe that no facility is closer than α*r (9.6) of a site and if a site has an existing facility within αr, all other sites are no closer than r (12) units away.

If one wished to examine the lower bound, as in the DACLP, one would have to solve the DACLP problem first, as this would provide the optimal minimally packed configuration. Then one would simply have to use the above formulation as well as a constraint that specifies that no more than p-facilities are deployed to ensure that a proper but maximal valued configuration would be found.

Figure 16. Generalized ACLP model optimal solution for a separation distance of 12 with α equal to 0.8 and all site benefits equal to 0 applied to the Swain dataset with 17 located facilities



2. The Site Sensitive *r*-Separation ACLP & DACLP

To illustrate the importance of using site-sensitive minimum separation standards, consider the following examples. For instance, one may be interested in modeling a territorial species that must sustain itself by defending an area centered about its nest/den. Let us also consider the habitat in which the animal resides; it should contain a source of water, food, and shelter that enable it to survive and thrive. One must recognize that areas distributed over space vary in habitat quality. For example an area might have a river that feeds into a lake with several species of fish in varying quantities as well as tree/shrub species that provide nesting material and/or food, while another area is centered over several creeks. The lake and river habitat has a greater capacity to provide fish (food) and nesting sites/material (trees/shrubs) than the area with many creeks. Both are suitable habitat locations, for example an osprey, but require two different territory sizes to maintain the same level of access to resources such as food and nesting material.

Another example of the importance of location related to territory could be a competitive retail chain. A competitive retail chain may locate stores with varying market sizes (varying *r*-Separations) based upon the underlying threshold of market support required to maintain the viability of a store. So then, if one is to estimate the number and distribution of a territorial species over an area or the location of competitive retail stores, should one not also consider the suitability of each site and its influence on the size of the separation standards? What would these distributions and capacities be? Before addressing these questions, however, the problem must be restated in a general form.

The Variable r-Separation Anti-Covering Problem: given several locations that vary in quality with constant threshold requirements, what would be the optimal arrangement of the nest/den/facility sites?

The required separation between neighboring nest/den/facility locations is a function of the threshold requirement at a potential site of location; for the territorial species it could be the quality of the underlying landscape, or the size and level of access to a customer base for a store. In the Variable r-Separation Anti-Covering model, the separation distance r for very high quality locations will be smaller than the r distance required in lower quality locations.

This assumption is supported in the territorial species literature, of which a nice review can be found in the paper by Joni Downs and Mark Horner (2008), which specifically looked at the effects of point patterns and the shape of home range estimates using spatial statistical methods. Justin Williams (2008) also considered reserve site selection with distance requirements. Church and Bell (1990) looked at the variations in business site location and their effect on Central Place Theory, particularly with hybrid k-levels in their paper. Varying radii in the Variable r-Sep ACLP capture these differences and could lead to such a landscape identified by Church and Bell. Thus, this is an important element of dispersion modeling that is missing and should be developed.

Carrying capacity estimates of habitat using the optimistic view ("Rosy View") of the ACLP have been generated; most notably and recently by Downs, Gates, and Murray (2008). However, their paper does not consider nesting site quality, and its impact on the value of r. The Variable r Anti-Covering Location Problem (VrACLP) considers each site and the underlying quality of that site and maximizes the number of facilities located. The notation is the same as has been previously defined, including the following:

r_j is the radius of separation criterion for location *j*

Using this notation, the V*r*ACLP can be formulated thusly as:

$$\text{Maximize} \sum_j x_j \tag{39}$$

s.t.

$$nx_j + \sum_{k \in N_k} x_k \le n, \text{ where } N_k = \{ j \mid \text{site } j \text{ is within } r_j \text{ of } k \} \tag{40}$$

$$x_j \in \{0,1\} \quad \text{for each} \quad j \tag{41}$$

The objective in this formulation (39) maximizes the number of facilities placed. The site location suitability measure is captured in the separation standard criterion, r_j, and is adjusted accordingly for each site *j*. Constraints (40) prevent sites that are closer than r_j distance from site *j* from being used when site *j* is selected. Constraints (41) restrict the facility site selection variables to binary integer values.

The Variable *r* Anti-Covering Location Problem can be re-formulated to consider the minimum number of facilities that can be located, the lower bound, as in the DACLP as well. The formulation of that problem, the Variable *r* Disruptive Anti-Covering Location Problem (V*r*DACLP) is defined with the following additional notation:

$N_i = \{ j \mid \text{site } j \text{ is within } r_i \text{ of } i \}$

$$\Theta_i = \{ j \mid \text{site } i \text{ is within } r_j \text{ of } j \}$$

V*r*DACLP:

$$\text{Minimize} \sum_j x_j \tag{42}$$

s.t.

$$nx_i + \sum_{k \in N_i} x_k \le n_i \quad \text{for each } i \tag{43}$$

$$\sum_{j \in \Theta_i} x_j \ge 1 \quad \text{for each } i \tag{44}$$

$$x_j \in \{0,1\} \quad \text{for each } j \tag{45}$$

The objective function (42) minimizes the number of facility/nest centers to be placed, while considering site suitability. The constraints (43) and (45) have the same function as those previously discussed for the V*r*ACLP. Because this form is designed to find the optimally “disruptive” packing configuration, following the DACLP, this necessitates the inclusion of constraints (44). One issue that needs to be addressed with this formulation is that, while the model will locate the minimum number of facility sites such that no site is closer than the specified standard and that there are no other sites that may be packed, the model will also lead to solutions that are “biased”. This “bias” is towards sites located in lower quality habitat/territories because the radius of separation is larger and site suitability is lower.

This means that a greater area can be covered by fewer, lower quality, nest/den/facility sites with greater separation distances when in reality there is likely to be a mix of high and low quality sites. A territorial species, and a business, will certainly tend to select higher quality sites. Depending on how great the variation between the optimistic and pessimistic views, it may be necessary to account for variations that lie between the "pessimistic" and "rosy" outlooks by attempting to generate solutions in between the two bounds. This could easily be done by adding constraints (26). Constraints (26) specify that at least *p*-facilities be located between the p_{min} (ACLP) and p_{max} (DACLP) solution configurations. In this way varying configurations could be generated. Using this constraint will assist in providing a range of configurations of possible outcomes.

C. Conclusions

The classic ACLP involves maximizing the number of facilities being placed while keeping them at least *r* distance apart from each other. The ACLP has been used in a number of different application areas, including reserve design, defense, forest operations models, DNA sequencing, analyzing policies impacting potential sex offender residence location as well as potential liquor store patterns, among several others. This chapter has presented several important extensions of the Anti-Covering Location Problem (ACLP).

Section A introduced a new problem called The Disruptive Anti-Covering Location Problem, which discussed the importance of the Disruptive Anti-Covering Location Problem (DACLP) in policy analysis. The disruptive anti-cover problem

(DACLP) involves finding a solution which minimizes the number of facilities being placed, ensuring that all facilities are separated from each other by a minimum separation distance, *r*, and where no further sites can be selected without violating a separation condition. An integer-linear formulation model for the disruptive anti-covering problem has been developed, example solutions provided, and a discussion of how this model can be used to identify stable levels of disruption was included.

In particular, this section demonstrates the importance of the disruptive form of anti-covering. In fact, when policies are analyzed using the anti-cover location model (*e.g*. sex offender residences or carrying capacity of a population of Sandhill cranes), it makes sense to solve the disruptive form of this problem as well in order to capture the range of possible outcomes. When the problem encompasses a number of independent decision making entities, sex offenders (or birds) in selecting housing (or nest sites), it is likely that an optimal pattern will not be generated. Thus, solutions to the DACLP are important and informative within the context of policy analysis and decision making.

A bi-level "leader and follower" model is also proposed for identifying placement strategies to thwart or disrupt optimal configurations to the greatest extent possible with limited resources. This bi-level model was only formulated; this model is a research avenue that should be explored in greater detail and is left for future work.

Section B. "Other Forms of r-Separation" focuses primarily on how one could consider variations in the way separation standards are imposed. All previous formulations in the literature are based on the assumption that there is a fixed separation standard. However, there are cases where such strict standards are not as representative as one would like. This is especially true where explicit standards of separation are not appropriate for certain modeling applications, as in models representing habitat carrying capacity for territorial species or where certain franchise stores or retail outlets share a portion of market area. The generalized ACLP model allows modest violations of separation constraints in selecting a configuration and encourages the use of higher-valued sites.

In addition to the Generalized ACLP, another problem called the Variable ACLP was proposed. This problem is based on the assumption that the separation standards could be site specific. One example of an application of this type of problem involves the analysis of carrying capacity. For those locations where resources are plentiful, separations between one individual and others is likely not as important as those locations that provide fewer benefits. Sites with lower benefit are likely to force an individual to maintain a larger territory to maintain a similar level of resource availability than a site with greater benefit. The Variable r-Separation ACLP was formulated as an integer linear programming problem and this should be the subject of future research.

Overall this chapter presents two new major conceptual models related to the ACLP have been developed. The first section pointed out that the ACLP is often "rosy" when used to determine optimal solutions to the ACLP and a related problem, the Disruptive ACLP, was discussed and formulated to address this issue. The subsequent sections discussed how separation standards themselves can be relaxed or specified individually. Each of these points are important to the various applications for which the ACLP has been employed. The next chapter, “Computational Results” presents the results of the computational experience in solving ACLP and DACLP models.

In conclusion, this chapter has focused on:

- Describing and formulating the Disruptive Anti-Covering Location Problem
- Defined proper solutions and discussed the importance of intermediate anti-covering solutions that range from the upper ACLP bound and DACLP lower bound
- Presented a bi-level leader and follower formulation for disruption
- Covered two ways of implementing a separation standard

V. Computational Results

As discussed in chapter III and IV, the Anti-Covering Location Problem (ACLP) and the Disruptive Anti-Covering Location Problem (DACLP) are Non-deterministic Polynomial-time Hard, or NP-Hard, problems. This means that large instances of these problems may be difficult to solve to provable optimality, even if given significant computing resources and time. For this reason several sophisticated constraint representations have been developed (e.g. Erkut & Neuman, 1991; Yoshimoto & Brodie, 1994; Erkut, ReVelle, & Ulkusal, 1996; Murray & Church, 1996; Murray & Church, 1997a) in addition to several heuristic approaches. This chapter is composed of two primary sections that present and compare computational tests of competing model formulations using off-the-shelf commercial solvers and a section that introduces a new heuristic approach.

The first section discusses the computational experience of solving the ACLP and DACLP to optimality using an industrial solver. Several of the modeling constraint structures discussed in chapter three are used to represent the ACLP and DACLP and are solved to optimality. By using industrial off-the-shelf software where each constraint structure is formulated, a comparison of each structure can be conducted and discussed. The second section contains a brief overview of existing heuristic solution approaches to solving the ACLP as well as a new heuristic approach called the Marching Army heuristic. Furthermore, a discussion on heuristic approaches to solve the DACLP is provided.

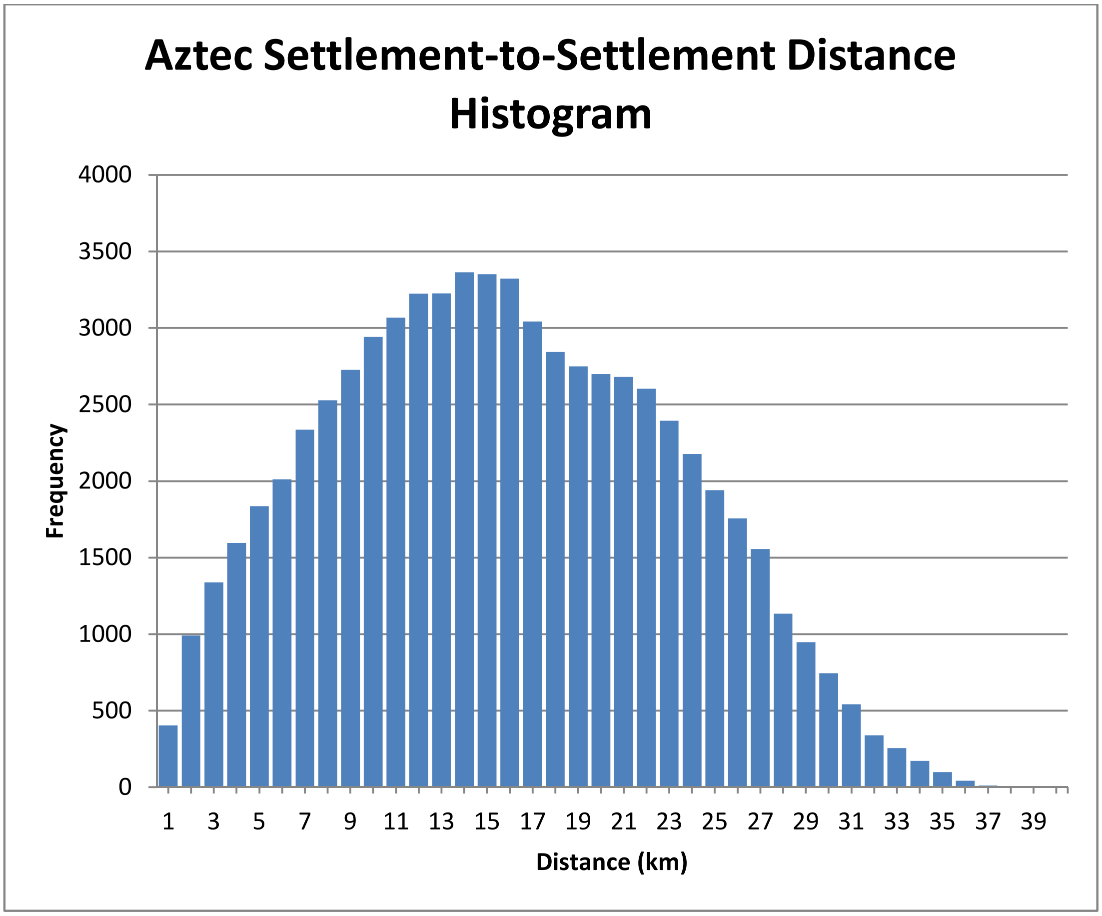
A. Solving the ACLP and DACLP to Optimality

Solving the ACLP and DACLP using most of the model forms presented in chapters 3 & 4 to optimality on small datasets can be done with ease. For example, many of the conceptual figures used in the previous chapters were solved to optimality in less than a tenth of a second using the Swain (1971) dataset. However, as the size of problems increase computation time often increases as well. This chapter section utilizes the larger dataset of Ruggles and Church (1996) to compare the various model formulations representing the ACLP and DACLP.

The Ruggles and Church (1996) dataset represents a collection of Aztec settlements. The settlements range in size from hamlets to cities and are dispersed over a large region of approximately 900 km^2. In total there are 372 point locations in the dataset. Each point represents the centroid of the town center. Figure 17 shows the histogram of the distance matrix representing the Euclidean settlement-to-settlement distances of the 372 Aztec settlements. The average settlement-to-settlement distance is 15 kilometers, with the bulk of the settlement distances between 5 and 25 kilometers. This is important in terms of thinking about separation standards. If the distance of separation standard, r, is small and there are few neighbors, the neighborhood constraints and cliques will be sparsely populated. As the value of r increases, the neighborhood constraints will increase in their membership and thus size. As constraint membership, be it clique or neighborhood constraint, population size and complexity typically increase as well. Thus, the

Ruggles and Church (1996) dataset is particularly well suited to testing separation standards.

Figure 17. Histogram of the settlement-to-settlement distances of the 372 point Aztec dataset of Ruggles and Church (1996)



The Ruggles and Church dataset was used as it is larger than the Swain (1971) dataset and is large enough and contains enough spatial complexity that computation times will measurably vary. This is important because the optimizer must be stressed by enough problem complexity to compare solution times. In order to solve each of the represented formulations, the Fair Isaac Corporation's (FICO) Xpress solver version 25.01.05 was used. FICO's Xpress-IVE development environment including the Mosel modeling language version 3.4.3 was used to set up each problem which was then solved using the Xpress solver. Mosel was used as it is very easy to formulate a model and check it, and Xpress is an industrial solver just as capable as CPLEX. Only the Core and Wedge and Maximal Clique formulations used specialized code. In this case the model forms were generated using a program developed in Microsoft's Visual Basic .Net environment. These models were then solved to optimality by means of the same Xpress solver. The primary reason for doing this was to take advantage of GIS functionality for the Core and Wedge geometry calculations, and to take advantage of the multi-thread capabilities in the computation of the vertex packing problems that are required in the maximal clique formulation of Murray and Church (1997a). Three primary computers were used to run the software.

The three different computers used to run the models were: 1) Super-Chief, a 2.8 GHz quad-core hyper-threaded[4] (8 total threads) Intel i7 CPU desktop computer with 12GB of PC3-10700 (1333 MHz) memory running the Windows 7 operating system (OS); 2) Jupiter, a 4.1GHz quad-core (4 total threads) AMD A-10 6800K CPU desktop computer with 8GB of PC3-17000 (2133 MHz) memory running the Windows 7 OS; and 3) IO, a 2.5GHz variable speed mobile quad-core (4 total threads) AMD A-10 5750M CPU laptop computer with 8GB of PC3-12800 (1600 MHz) memory running the Windows 8.1 OS. Super-Chief is the computer for which

[4] A hyper-threaded CPU means that there is a "virtual" core associated with an individual physical core. For example a dual-core hyper-threaded processor has 4 threads to which computation may be distributed. The threads of a CPU are where computation is conducted for a process.

the results of several model formulations were run using a range of separation distances.

A side interest in running the various models on these configurations was to determine if the hyper-threading capability of the Intel processor was of benefit when running the models using the Xpress solver. The other computers were used as their CPU configurations represent a broad range of hardware from which performance can be compared. Super-Chief has the benefit of multiple threads per core, albeit at a reduced CPU clock frequency, whereas Jupiter has the advantage of a higher clock speed but the disadvantage of having fewer threads. IO is used to compare a more efficient mobile CPU to the desktop CPUs. The next two sub-sections report the results of the computational experience for the various formulations for the ACLP and DACLP problems entirely run on SuperChief. Following that, a comparison is made across all of the computer configurations.

1. Computational Experience of Various Constraint Forms Representing the Anti-Covering Location Problem

This sub-section presents the computational experience for the various constraint representations of the ACLP. All of the results in this section were generated using Super-Chief, a computer with a 2.8 GHz quad-core i7 CPU and DDR3 memory described previously. Before the results are described, a brief review of the varying constraint structures are characterized using the following notation:

i, j are indices of potential facility locations

r is the minimum distance standard, or radius of separation

S is the set of potential facility site locations

Q_i $= \{ j \in S \mid d_{ij} < r \quad \text{where} \quad j \neq i \}$, defined for each $i \in S$

n_i is the largest number of sites which can be simultaneously selected within the set Q_i while maintaining a distance separation of r between each pair of facilities

d_{ij} shortest distance from facility i to facility j

x_j decision variable where $\begin{cases} 1, \text{if facility is sited at } j \\ 0, \text{otherwise} \end{cases}$

Using this notation, the following generic neighborhood constraint represents the restrictions on siting facilities by ensuring that if site *i* is chosen for a facility, then all other sites that are within the *r* standard (distance) of location *i* must be left unused. The generic neighborhood constraint can be written as:

$$n_i x_i + \sum_{j \in Q_i} x_j \leq n_i \qquad \text{for all } i \in S \tag{46}$$

This is the neighborhood constraint first proposed by Yoshimoto and Brodie (1994), though they had $|Q_i|$ including site *i* instead of n_i. Murray and Church (1995) recognized that $|Q_i|$ could be further reduced to the form of n_i. Representations using n_i are an improvement in that they are much more efficient than that of the big *M* value[5] implemented by Moon and Chaudhry (1984). In addition to the neighborhood constraint form, the ACLP can be represented as a set

[5] A very large integer number, such as 999,999.

of pairwise constraints, as well as a hybrid form composed of a neighborhood constraint and clique constraint.

A pairwise constraint can be viewed in the following way. For example, consider three facility sites *t*, *u*, and *v* which are within *r*-distance of each other. In pairwise form they are represented as: $x_t + x_u \leq 1$, $x_u + x_v \leq 1$, and $x_t + x_v \leq 1$. These three constraints can be reduced into one inequality term: $x_t + x_u + x_v \leq 1$. This reduced representation is called a clique. As stated in the cliques section of chapter three, it is sensible to combine pairwise constraints whenever possible into higher ordered clique sets. This reduces the needed number of constraints and produces a tighter relaxed problem. In general a clique constraint can be written as:

$$x_i + \sum_{j \in C} x_j \leq 1 \tag{47}$$

Where *C* is the set of members of the neighborhood clique of facility site *i*. The ACLP can be represented as: 1) using neighborhood constraints of type (46) such as: the Big-M and n_i forms – the total number of neighbors within *r* of facility site *i*; 2) A combination of neighborhood and clique constraints of types similar to (46) & (47) such as: Core Cliques, Maximal Cliques, and Pairwise forms; or 3) Entirely by cliques of similar form to type (47) such as Core & Wedge[6].

[6] In chapter three of this dissertation all of these constraint forms are discussed in detail. Big-*M* is the classic Moon and Chaudhry (1984) formulation, n_i is the tighter

Table 4 contains the computational experience of formulating and solving each of the constraint representations for the Anti-Covering Location Problem (ACLP). This table excludes the use of Location Set Covering Problem (LSCP) facet enhancing constraints, which will be discussed a little later in this sub-section. Table 4 presents each modeled form by row. The table is broken into two sections; the top half represents *r* separation standards of 1-10 in increments of 1. The bottom half represents *r* separation standards of 11-15 in increments of 1 and 20, 25, 30, 35, & 40. The columns represent each of the separation standards used to run each constraint form. The top rows of each section contain the separation distances used for each modeled constraint form. The second rows of each section contain the optimal ACLP objective value that was obtained for each distance standard, *r*. The rows associated with a particular model form contain the following: setup time, solution time, and total time. All times are reported in seconds. Results related to each formulation will be discussed independently first, and then compared and contrasted as a whole.

The Big-M approach (Moon & Chaudhry, 1984) is represented by a set of neighborhood constraints similar to constraints of type (46), as previously discussed. Observe in Table 4, that the Big *M* model form for a distance of separation of 5 required .02 seconds to set up and 2.7 seconds to solve, resulting in a total time of 2.72 seconds. When the model is solved to optimality, particularly for small *r*, it is

---

form proposed by Yoshimoto and Brodie (1994) and Murray and Church (1995), Core-Cliques by Erkut et al. (1996), pairwise constraints and maximal cliques described by Murray and Church (1997a; 1997b), and the completely new all clique constraint representation, Core and Wedge, also described in chapter three.

solved in less than a second. This is because when *r* values are small the total number of neighborhood members is quite small which results in a fairly compact constraint that is easily solved. As the *r* standard increases the members of each neighborhood steadily increase. As these neighborhood sizes increase the computational effort required to solve the problem increases. When the *r* standard is less than or equal to half of the average settlement-to-settlement distance (15km), the number of sites with overlapping neighborhoods is much more manageable and results in an easier to solve problem.

Once the average settlement-to-settlement distance is exceeded by the *r* separation standard the number of sites that are members of overlapping neighborhoods greatly increases. This results in several constraints with very large neighborhood membership that also overlaps, which increases the complexity of a given problem. This is reflected in the total solution time for a given problem as problem size increases. For example, it takes 6.077 seconds to solve for a standard of 15km. For a standard of 20km, the total required solution time takes over a minute (74.135 seconds). However, there is a way to further tighten a formulation with using only neighborhood constraints.

This brings us to the Yoshimoto and Brodie (1994) constraint representation. Instead of "Big-*M*", this form utilizes $|Q_i|$ in each neighborhood constraint. The performance of the Yoshimoto and Brodie constraint representation is very similar to the Big-*M* representation (see Table 4: Yoshimoto & Brodie). In many instances this formulation is solved in almost the same, or in slightly less, time than the Big-*M* representation.

Table 4: Computational results associated with solving 6 ACLP model formulations applied to the 372 node Aztec data set. All times are in seconds and *r*-Sep in kilometers

	r-Sep	1	2	3	4	5	6	7	8	9	10
	Objective	212	110	68	45	32	26	21	17	14	11
Big M	Setup Time	0.017	0.014	0.019	0.019	0.020	0.019	0.020	0.021	0.023	0.023
	Solution Time	0.035	0.069	0.212	0.882	2.700	1.453	1.904	2.949	2.266	2.284
	Total Time	0.052	0.083	0.231	0.901	2.720	1.472	1.924	2.970	2.289	2.307
Yoshimoto & Brodie	Setup Time	0.016	0.018	0.017	0.018	0.018	0.019	0.020	0.021	0.022	0.023
	Solution Time	0.034	0.069	0.214	0.823	2.901	1.320	1.825	2.897	2.259	2.270
	Total Time	0.050	0.087	0.231	0.841	2.919	1.339	1.845	2.918	2.281	2.293
Pairwise	Setup Time	0.022	0.024	0.025	0.026	0.028	0.031	0.031	0.034	0.037	0.039
	Solution Time	0.037	0.075	0.124	0.273	0.827	0.521	0.784	1.325	1.211	3.880
	Total Time	0.059	0.099	0.149	0.299	0.855	0.552	0.815	1.359	1.248	3.919
Core Cliques	Setup Time	0.027	0.028	0.028	0.028	0.030	0.031	0.032	0.033	0.034	0.036
	Solution Time	0.038	0.079	0.247	0.706	3.383	1.279	3.155	3.971	2.105	4.457
	Total Time	0.065	0.107	0.275	0.734	3.413	1.310	3.187	4.004	2.139	4.493
Core & Wedge Cliques	Setup Time	0.030	0.030	0.070	0.030	0.030	0.030	0.040	0.040	0.040	0.040
	Solution Time	0.000	0.000	0.000	0.100	1.400	0.300	1.000	1.100	0.500	0.800
	Total Time	0.030	0.030	0.070	0.130	1.430	0.330	1.040	1.140	0.540	0.840
Maximal Cliques	Setup Time	19.210	19.010	18.710	18.520	19.600	18.880	19.860	19.410	20.260	23.410
	Solution Time	0.000	0.000	0.100	0.200	1.200	0.700	1.400	2.600	1.100	2.200
	Total Time	19.210	19.010	18.810	18.720	20.800	19.580	21.260	22.010	21.360	25.610

	r-Sep	11	12	13	14	15	20	25	30	35	40
	Objective	10	9	8	8	6	4	3	2	2	1
Big M	Setup Time	0.024	0.024	0.026	0.027	0.029	0.035	0.036	0.038	0.042	0.038
	Solution Time	2.688	2.924	3.765	4.688	6.048	74.100	79.900	88.200	78.800	84.400
	Total Time	2.712	2.948	3.791	4.715	6.077	74.135	79.936	88.238	78.842	84.438
Yoshimoto & Brodie	Setup Time	0.024	0.025	0.026	0.027	0.028	0.032	0.035	0.039	0.038	0.037
	Solution Time	2.636	3.006	3.793	4.723	6.255	73.100	79.800	88.220	78.500	84.690
	Total Time	2.660	3.031	3.819	4.750	6.283	73.132	79.835	88.259	78.538	84.727
Pairwise	Setup Time	0.041	0.053	0.046	0.051	0.053	0.065	0.075	0.080	0.082	0.080
	Solution Time	3.522	4.940	4.619	5.261	6.401	12.300	52.300	138.000	36.000	111.000
	Total Time	3.563	4.993	4.665	5.312	6.454	12.365	52.375	138.080	36.082	111.080
Core Cliques	Setup Time	0.038	0.038	0.040	0.042	0.043	0.052	0.055	0.057	0.058	0.057
	Solution Time	1.520	0.526	1.553	0.943	0.983	1.538	4.243	18.800	35.300	25.100
	Total Time	1.558	0.564	1.593	0.985	1.026	1.590	4.298	18.857	35.358	25.157
Core & Wedge Cliques	Setup Time	0.040	0.040	0.040	0.050	0.050	0.060	0.060	0.060	0.070	0.060
	Solution Time	0.800	1.000	1.400	1.600	1.900	3.200	6.400	24.100	40.500	48.800
	Total Time	0.840	1.040	1.440	1.650	1.950	3.260	6.460	24.160	40.570	48.860
Maximal Cliques	Setup Time	28.850	39.160	40.260	50.560	58.820	52.540	39.490	19.650	19.350	19.250
	Solution Time	1.900	1.400	0.300	0.400	0.700	0.600	0.600	0.100	0.000	0.000
	Total Time	30.750	40.560	40.560	50.960	59.520	53.140	40.090	19.750	19.350	19.250

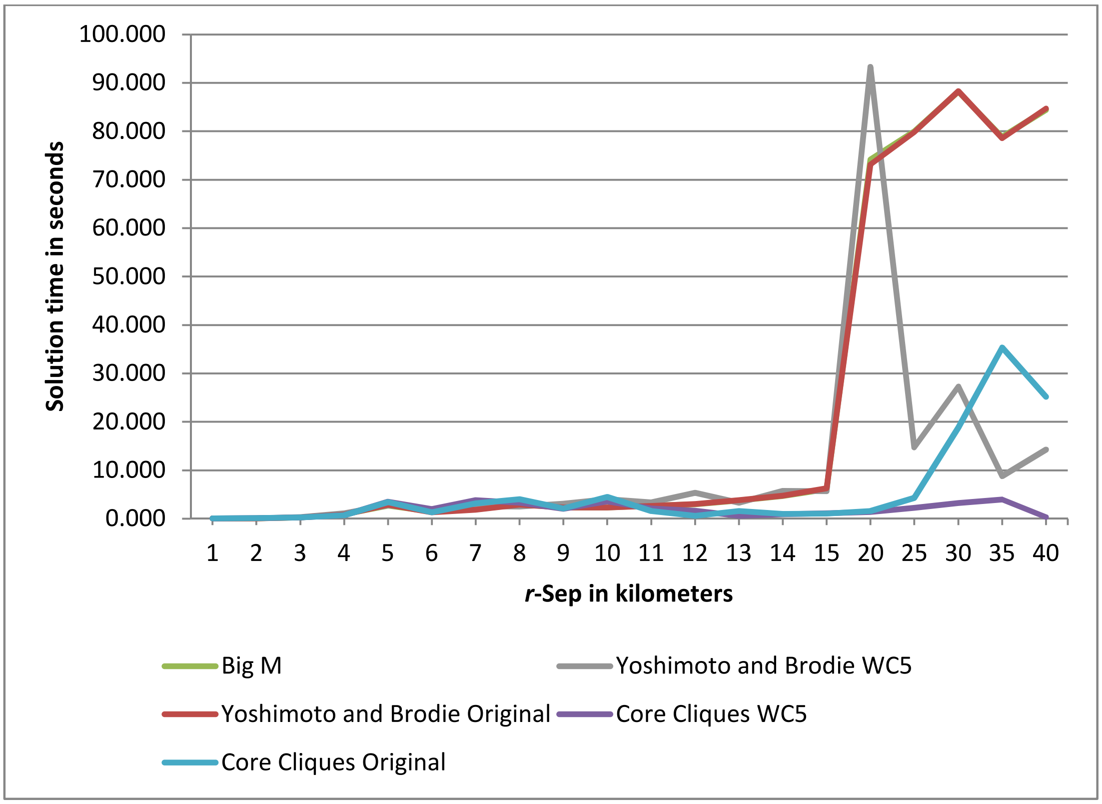
Since this approach is functionally similar to the Big-*M* approach, it is not surprising that the performance is similar, to slightly better. One of the reasons why this approach is a slight improvement over that of Big-*M* is because the number of members in the neighborhood set is still quite large using Yoshimoto and Brodie's approach.

Because of this, the model still suffers from the same problem as that of the Big-*M* approach. As the number of members of a neighborhood increase and as neighborhood overlap increases, the computational effort required to solve the problem increases as well. This, again, is especially apparent when separation standards exceed the average settlement-to-settlement distances. For example when the *r* standard is 15km it takes 6.283 seconds to solve and when *r* is 20, it requires 73.132 seconds to solve.

Given that total solution times only varied slightly or subtly improved using Yoshimoto and Brodie's approach, one might wonder how the proof that n_i is no larger than five would affect performance. The Wedge and Core approach, described in chapter three, led to the proof that the value of n_i need be no larger than 5. Thus, from the theoretical results of chapter 3, one can set n_i to five whenever the neighborhood size is larger than 5. Figure 18 graphically shows the total solution times for each neighborhood model form using a variety of separation standards.

Figure 18 shows that even though the neighborhood constraints can be tightened, using a maximum value of 5 as compared to the Big-*M* and Yoshimoto and Brodie the models perform similarly. Where the separation standard is less than the average settlement-to-settlement distance, the models solution behavior is almost indistinguishable. Where there is a greater difference in solution times is when the separation standard exceeds the settlement-to-settlement average distance. In this case, the tightened neighborhood constraint following the Wedge and Core proof generally leads to significant reductions in computation time required to solve a model.

Figure 18. Graphical example of neighborhood constraint representation total solution times



The Pairwise constraint representation requires very little computation time when the separation standard is small and the number of site locations within the separation standard of a given site are few in number. Since there are fewer sites within the separation standard of a given site, the overall size of the problem remains small. In addition, these constraints are considered to be tight. This leads to low computation times when the separation standard is small. In fact, the Pairwise formulation when solved for small separation standards is quite fast; when *r* is less than 9 km total solution times were computed in less than 1.5 seconds (see Table 4: Pairwise). However, as the separation standard increases and the number of sites within the separation standard of a given site increase, the size of the problem – that is the total number of constraints, dramatically increases which requires the solver to work harder in resolving fractional solutions. This is especially true when solving for large values of *r*. In fact, the largest required solution time in Table 4 is associated with the pairwise formulation ($r = 30$). Since several pairwise constraints may be represented as a single clique constraint, the performance of the clique-based models is reported next.

The first clique based formulation to be reported is that of Erkut et al. (1996) who describe a Core Clique that represents all site-site restrictions within half of the separation standard and a neighborhood constraint that represents those pairwise conditions restrictions for distances greater than half *r* distance to strictly within *r* distance away. The solution times of this approach when the separation standards are small, $r < 5$, total solution times are less than 0.75 seconds (see Table 4: Core Cliques). This is because the problem size is still relatively small and membership in neighborhoods and cliques is relatively low in number. When *r* is larger, the membership of the core cliques and neighborhoods grow larger, and so does needed computational times. Overall, total solution times are relatively low in comparison to other approaches.

When the separation standard is larger (11 to 15km) total solution times are much less, ranging from approximately 0.5 to 1.5 seconds than for distances of 5 to 10. When the separation standard exceeds the average settlement-to-settlement distance, computation time increases, though not as dramatically as in the previous approaches. At very large separation standard values (20-40km), computation times range from approximately1.5 to 35.5 seconds. The reason for the relatively lower computation times at larger separation standards is that the core clique membership becomes larger and represents spatially a large percentage of pairwise restrictions. This happens even as the neighborhood membership remains relatively constant. This makes for a fairly tight and compact problem, which is reflected in the computation times, particularly for larger separation standards.

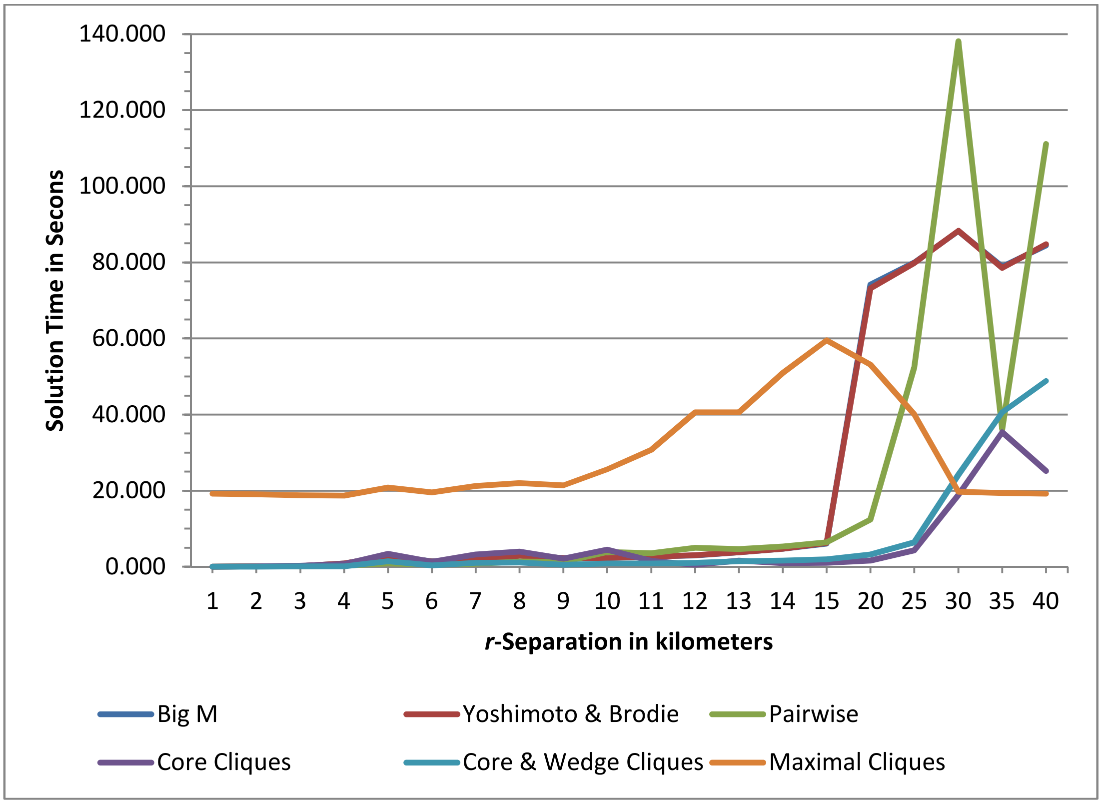
The Core and Wedge approach is based on the fact that an ACLP model in Euclidean space could be built entirely of at most 7 cliques per site. In this case, 7 clique constraints need to be written for a potential facility location unless a given clique for site *i* contains only site *i* as a member. Because cliques are very tight problem representations, models that use them are generally very efficiently solved. The problem size is much larger than that of a problem using a single neighborhood constraint set (*e.g* Yoshimoto and Brodie, 1994) and can take longer to compute a solution unless the separation distances are large enough to create neighborhood sets of relatively large size.

When separation standards are small (<5km) total solution times for the Core & Wedge model are less than 0.1 seconds. When separation standards are in the intermediate range (5 to 10km), total solution times are a little longer; they range from approximately 0.3 to 1.40 seconds. However, when separation standards are larger than 10 and less than 25 the approach is very efficient. Total solution times at larger separation standards range from approximately 0.24 seconds to 49 seconds.

The Maximal Clique model of Murray and Church creates a single very tight clique constraint for each location site and a neighborhood set that represents the remaining site locations not in the clique. This approach requires several vertex packing problems to be solved for each site neighborhood in order to generate the appropriate maximal clique to represent the problem. The process of generating each maximal clique was multithreaded, that is split into several sub-problems and run on all available CPU threads, to reduce problem set up times. However, even when the setup process is multi-threaded the process overall takes a great deal of time.

In Table 4, notice that the set up times for Maximal Cliques are large (always greater than 18 seconds). Using this approach results in very small solution times for the model, as solution times range from less than one thousandth of a second to a maximum of 2.2 seconds. This is because the formulation is very tight and easy to solve. However, the approach results in an overall total solution time that is very long; total solution times range from 18.720 seconds to 59.520 seconds. Unfortunately, solving the required vertex packing problems to set up the problem takes time. Thus, while the formulation problem is very efficient to solve to optimality, the setup process is not.

Figure 19. Total solution times of the various constraint representations of the Anti-Covering Location Problem



Given each of these formulation types, one may wonder which formulation approach may be best geared toward solving an Anti-Covering Location Problem (ACLP). Figure 19 presents the total solution times for each approach over the range of separation distances. The Core & Wedge formulation and the core formulation have robust performance when compared to the other formulations. Both solve relatively quickly and often outperform many of the other formulations, particularly for large separation standards. The Core and Wedge formulation outperforms the core model formulation when the separation standard is of small to intermediate size. This is because several tight cliques result in a fairly tight overall problem, whereas the Erkut et al. (1996) formulation does not.

When distances are very large and few facilities can be placed among the sites without violating a distance standard, the core model of Erkut et al. (1996) has the advantage over the core & wedge model. This is due to the fact that the core & wedge has 3.5 times more constraints. This is because for each site the core constraint overlaps with many of the other facility sites. This results in a relatively compact model where the core constraint acts like a maximal clique constraint and solves efficiently. Thus, the Core and Wedge and core modeling approaches are likely to be the most robust in terms of overall solution time; Core and Wedge performs the best for small to intermediate separation standards while the Core performs the best for intermediate to large separation standards. There is, however an additional constraint, the Location Set Covering Problem constraint, which may be added to all of these formulations with the potential to reduce solution times. This is discussed in the next subsection.

2. Computational Experience of Anti-Covering Location Problems with added Location Set Covering Constraints

The Location Set Covering Problem (Toregas & ReVelle, 1972), or LSCP, constraint could be added as a constraint to each of the previously described formulations. The LSCP constraint can be added in an attempt to create an efficient cut for the problem matrix. In other words, this constraint was proposed as a possible method to create facets which yield integer solutions with greater frequency than otherwise would occur. The LSCP constraint, using this added notation $N_i = \{j \in S \mid d_{ij} < r\}$ for all i in S, has the following general form:

$$\sum_{j \in N_i} x_j \geq 1 \qquad \forall \quad i \in S \tag{48}$$

For each ACLP model formulation an LSCP constraint set was added and solved for each problem instance to optimality.

Table 5 presents the computational experience results for each formulation with the appended LSCP constraints. In many cases where the separation standard was small the addition of the LSCP constraints resulted in a total solution time that was the same or slightly worse. This makes sense as the added constraint set increases problem size for problems that are already generally easily solved by the optimizer. However, where the LSCP constraints generally improve the total solution time performance is when the separation standard is larger than the average settlement-to-settlement distance. This is because there are many members of cliques or neighborhood constraints, and the addition of the LSCP constraint often results in a tight cut to the problem matrix.

Figure 20 shows this graphically. Results of the total solution time with and without the LSCP constraints for each formulation are presented. It is clear that when separation standards are small, the computation is increased through their addition. However, for large separation standards, there is a general reduction in solution time for the pairwise and core models. Solution times are also differentially affected

Table 5: Computational Experience of Formulations with added LSCP constraint

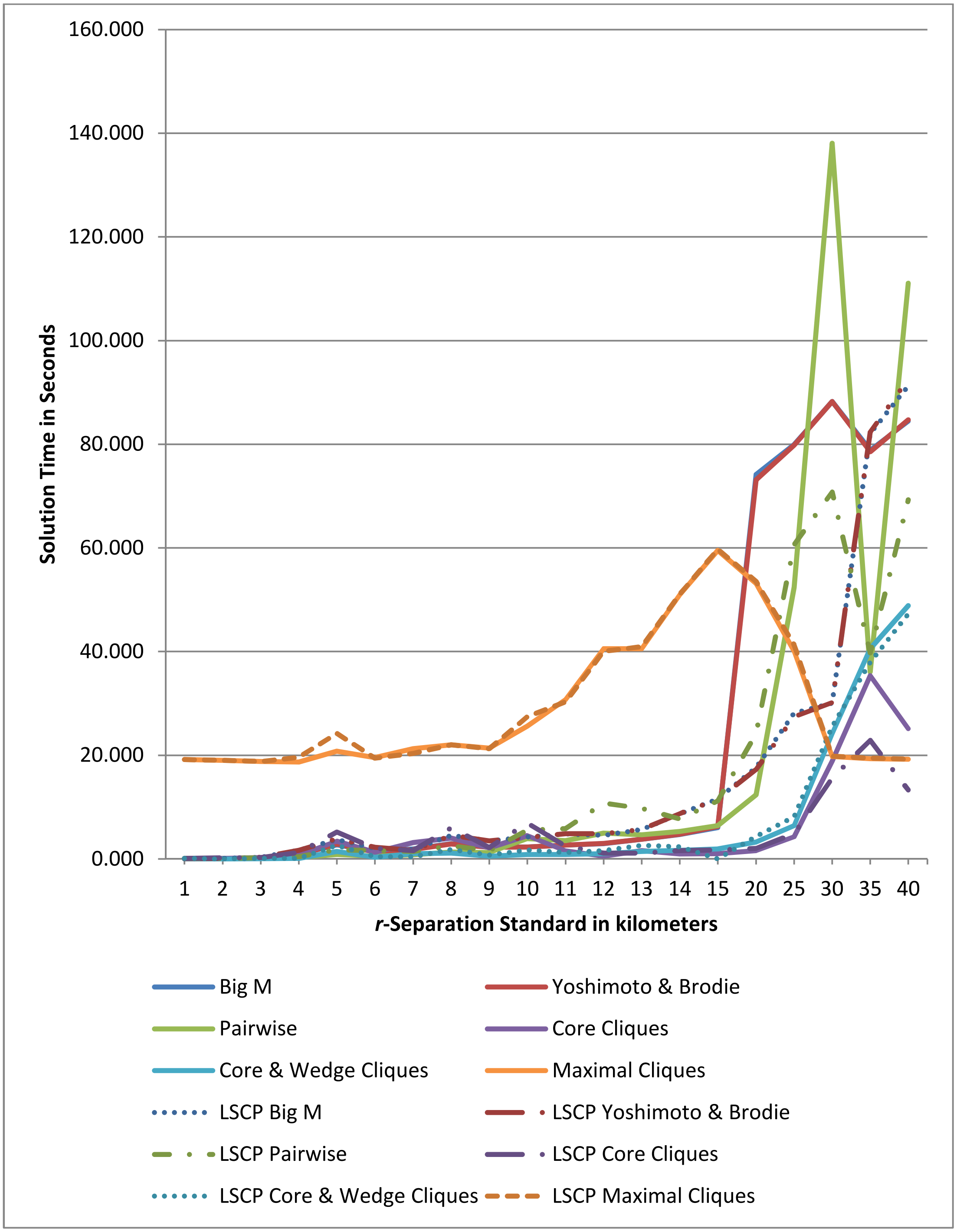
	r-Sep	1	2	3	4	5	6	7	8	9	10
	Objective	212	110	68	45	32	26	21	17	14	11
Big M	Setup Time	0.026	0.026	0.027	0.027	0.041	0.029	0.034	0.037	0.033	0.033
	Solution Time	0.044	0.093	0.305	1.624	3.724	2.038	1.808	4.510	3.431	4.229
	Total Time	0.070	0.119	0.332	1.651	3.765	2.067	1.842	4.547	3.464	4.262
Yoshimoto & Brodie	Setup Time	0.026	0.032	0.027	0.027	0.029	0.036	0.030	0.032	0.034	0.035
	Solution Time	0.044	0.096	0.307	1.631	3.883	2.215	1.639	4.519	3.447	4.095
	Total Time	0.070	0.128	0.334	1.658	3.912	2.251	1.669	4.551	3.481	4.130
Pairwise	Setup Time	0.039	0.032	0.046	0.035	0.040	0.068	0.045	0.050	0.048	0.050
	Solution Time	0.059	0.085	0.169	0.396	1.720	1.520	1.414	2.467	1.560	5.541
	Total Time	0.098	0.117	0.215	0.431	1.760	1.588	1.459	2.517	1.608	5.591
Core Cliques	Setup Time	0.047	0.048	0.036	0.038	0.039	0.040	0.061	0.044	0.055	0.048
	Solution Time	0.049	0.197	0.262	1.216	5.162	2.019	1.579	5.971	2.135	6.869
	Total Time	0.096	0.245	0.298	1.254	5.201	2.059	1.640	6.015	2.190	6.917
Core & Wedge Cliques	Setup Time	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.040	0.040	0.040
	Solution Time	0.100	0.000	0.100	0.100	2.600	0.400	0.400	1.800	0.700	1.700
	Total Time	0.130	0.030	0.130	0.130	2.630	0.430	0.430	1.840	0.740	1.740
Maximal Cliques	Setup Time	19.210	19.010	18.710	18.520	19.600	18.880	19.860	19.410	20.260	23.410
	Solution Time	0.000	0.000	0.100	1.100	4.600	0.600	0.500	2.600	1.000	4.000
	Total Time	19.210	19.010	18.810	19.620	24.200	19.480	20.360	22.010	21.260	27.410

	r-Sep	11	12	13	14	15	20	25	30	35	40
	Objective	10	9	8	8	6	4	3	2	2	1
Big M	Setup Time	0.035	0.037	0.039	0.040	0.041	0.048	0.054	0.054	0.054	0.057
	Solution Time	4.799	4.522	5.628	8.644	11.540	17.600	28.100	30.000	82.000	91.200
	Total Time	4.834	4.559	5.667	8.684	11.581	17.648	28.154	30.054	82.054	91.257
Yoshimoto & Brodie	Setup Time	0.038	0.056	0.038	0.043	0.042	0.049	0.052	0.055	0.055	0.056
	Solution Time	4.825	4.849	5.633	8.675	11.500	17.200	27.400	30.100	82.300	92.500
	Total Time	4.863	4.905	5.671	8.718	11.542	17.249	27.452	30.155	82.355	92.556
Pairwise	Setup Time	0.057	0.082	0.071	0.064	0.085	0.084	0.091	0.097	0.102	0.096
	Solution Time	5.777	10.700	9.712	7.672	11.100	24.100	60.550	70.700	39.670	69.200
	Total Time	5.834	10.782	9.783	7.736	11.185	24.184	60.641	70.797	39.772	69.296
Core Cliques	Setup Time	0.049	0.055	0.053	0.059	0.058	0.065	0.071	0.075	0.074	0.075
	Solution Time	2.437	1.077	1.071	1.556	1.685	1.995	4.787	15.530	22.800	13.200
	Total Time	2.486	1.132	1.124	1.615	1.743	2.060	4.858	15.605	22.874	13.275
Core & Wedge Cliques	Setup Time	0.040	0.040	0.050	0.050	0.050	0.060	0.060	0.060	0.060	0.060
	Solution Time	1.200	1.600	2.600	2.300	2.400	4.300	8.100	25.500	37.900	47.200
	Total Time	1.240	1.640	2.650	2.350	0.030	4.360	8.160	25.560	37.960	47.260
Maximal Cliques	Setup Time	28.850	39.160	40.260	50.560	58.820	52.540	39.490	19.650	19.350	19.250
	Solution Time	1.500	0.900	0.700	0.500	0.900	1.000	1.800	0.200	0.100	0.000
	Total Time	30.350	40.060	40.960	51.060	59.720	53.540	41.290	19.850	19.450	19.250

depending on the formulation type. For example, the total solution times of pairwise constraints for large separation standards are greatly reduced, as are the Yoshimoto and Brodie and Big *M* formulations. The formulations that appear to receive less of a benefit when used in conjunction with LSCP constraints appear to be models based upon clique constraints, however, the core formulation appears to receive the greatest benefit when used with LSCP constraints.

Thus, it appears that LSCP constraints are useful, though only when there are larger separation standards. In conclusion, this sub-section has presented and discussed the results of several constraint formulations representing the ACLP. It appears that the Core and Wedge formulation and core models are quite efficient as compared to the other approaches. The core & wedge model outperforms the core model when used to solve small problems using small to intermediate separation standards. However, the core & wedge model is not as efficient as the core model when very large separation standards are used and only a few facilities can be located. The next section discusses the computational experience of solving the Disruptive Anti-Covering Location Problem.

Figure 20. Graphical comparison of solution times for formulations with and without the addition of Location Set Covering Problem Constraints



3. Computational Experience of Various Constraint Forms Representing the Disruptive Anti-Covering Location Problem

This section describes the computational experience of solving the Disruptive Anti-Covering Location Problem. The disruptive anti-covering location problem is presented in detail in chapter four. Essentially, this problem may be modeled using any of the ACLP formulations by adding LSCP constraints and changing the maximization objective to a minimization objective. Thus, if one develops an ACLP formulation with LSCP constraints one simply needs to switch the objective function and one has a working DACLP formulation and vice-versa. This is due to the fact that the LSCP constraints along with the separation constraints ensure that each solution is proper. Table 6 presents the results of the DACLP formulations computational experience.

The results of Table 6 are very similar to the results of the ACLP formulations in terms of the conclusions made about constraint membership and problem size for each formulation type. What *is* different about these results, however, is that all of the formulations representing the DACLP are solved to optimality significantly faster than their ACLP counterparts with and without LSCP constraints. There is a reason for this.

That reason is simply how cuts can be made to the problem matrix and how the solver works to solve each of these problems. As this is actually a conditioned LSCP problem, it is likely that this form is easier to solve. In this case, the LSCP constraints represent particularly tight cuts for the DACLP problem.

Table 6: Computational experience of formulations representing the Disruptive Anti-Covering Location Problem

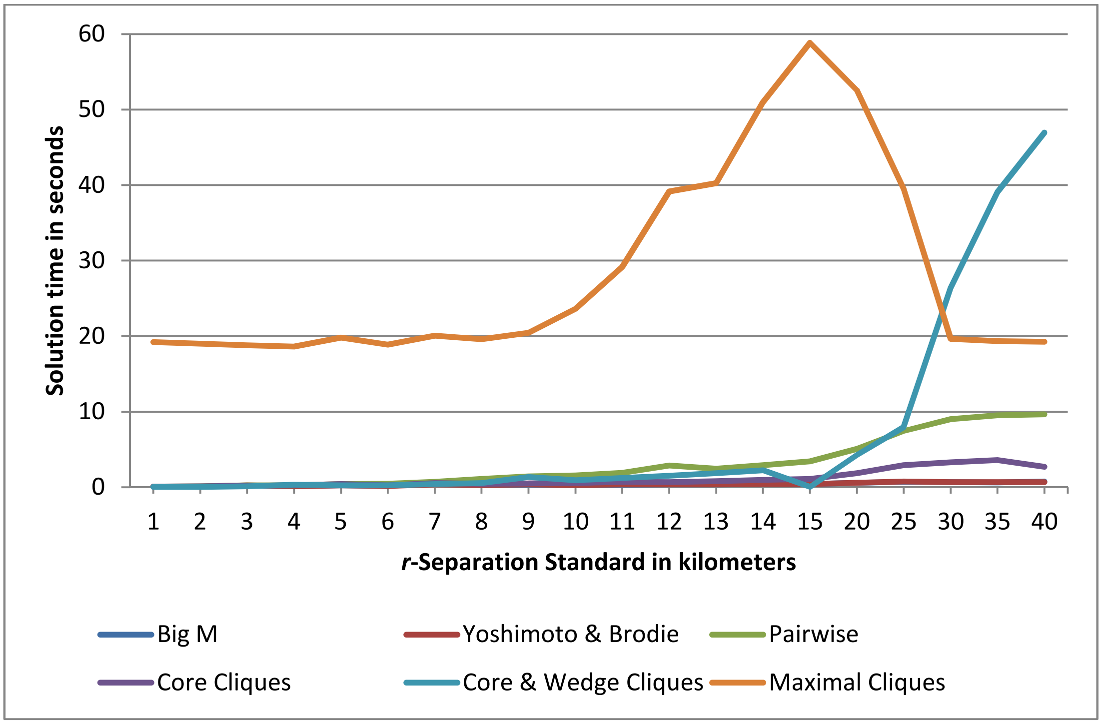
	r-Sep	1	2	3	4	5	6	7	8	9	10
	Objective	170	68	35	22	16	11	9	7	6	4
Big M	Setup Time	0.025	0.026	0.026	0.027	0.035	0.030	0.050	0.031	0.033	0.034
	Solution Time	0.046	0.061	0.164	0.101	0.213	0.143	0.484	0.246	0.317	0.278
	Total Time	0.071	0.087	0.190	0.128	0.248	0.173	0.534	0.277	0.350	0.312
Yoshimoto & Brodie	Setup Time	0.025	0.026	0.045	0.027	0.034	0.029	0.031	0.033	0.032	0.034
	Solution Time	0.040	0.065	0.184	0.099	0.228	0.145	0.365	0.261	0.330	0.275
	Total Time	0.065	0.091	0.229	0.126	0.262	0.174	0.396	0.294	0.362	0.309
Pairwise	Setup Time	0.031	0.032	0.035	0.036	0.037	0.039	0.041	0.049	0.047	0.050
	Solution Time	0.044	0.100	0.211	0.222	0.361	0.439	0.670	1.048	1.394	1.512
	Total Time	0.075	0.132	0.246	0.258	0.398	0.478	0.711	1.097	1.441	1.562
Core Cliques	Setup Time	0.045	0.046	0.046	0.059	0.050	0.052	0.054	0.062	0.058	0.060
	Solution Time	0.048	0.075	0.174	0.202	0.389	0.207	0.480	0.408	0.445	0.481
	Total Time	0.093	0.121	0.220	0.261	0.439	0.259	0.534	0.470	0.503	0.541
Core & Wedge Cliques	Setup Time	0.030	0.030	0.030	0.030	0.030	0.030	0.030	0.040	0.040	0.040
	Solution Time	0.000	0.000	0.100	0.300	0.200	0.200	0.400	0.500	1.300	0.900
	Total Time	0.030	0.030	0.130	0.330	0.230	0.230	0.430	0.540	1.340	0.940
Maximal Cliques	Setup Time	19.210	19.010	18.710	18.520	19.600	18.880	19.860	19.410	20.260	23.410
	Solution Time	0.000	0.000	0.100	0.100	0.200	0.000	0.200	0.200	0.200	0.200
	Total Time	19.210	19.010	18.810	18.620	19.800	18.880	20.060	19.610	20.460	23.610

	r-Sep	11	12	13	14	15	20	25	30	35	40
	Objective	4	4	4	3	3	1	1	1	1	1
Big M	Setup Time	0.036	0.038	0.039	0.040	0.041	0.048	0.052	0.054	0.054	0.080
	Solution Time	0.307	0.320	0.353	0.390	0.389	0.522	0.659	0.609	0.593	0.682
	Total Time	0.343	0.358	0.392	0.430	0.430	0.570	0.711	0.663	0.647	0.762
Yoshimoto & Brodie	Setup Time	0.039	0.043	0.038	0.039	0.045	0.047	0.058	0.054	0.055	0.055
	Solution Time	0.317	0.336	0.346	0.376	0.384	0.520	0.717	0.629	0.607	0.619
	Total Time	0.356	0.379	0.384	0.415	0.429	0.567	0.775	0.683	0.662	0.674
Pairwise	Setup Time	0.052	0.055	0.058	0.066	0.074	0.080	0.091	0.100	0.100	0.097
	Solution Time	1.848	2.810	2.394	2.840	3.366	4.981	7.336	8.917	9.409	9.539
	Total Time	1.900	2.865	2.452	2.906	3.440	5.061	7.427	9.017	9.509	9.636
Core Cliques	Setup Time	0.081	0.066	0.071	0.072	0.077	0.086	0.095	0.100	0.102	0.103
	Solution Time	0.627	0.610	0.739	0.889	1.020	1.786	2.838	3.208	3.490	2.590
	Total Time	0.708	0.676	0.810	0.961	1.097	1.872	2.933	3.308	3.592	2.693
Core & Wedge Cliques	Setup Time	0.040	0.040	0.050	0.050	0.050	0.060	0.060	0.060	0.060	0.060
	Solution Time	1.200	1.500	1.800	2.200	2.400	4.200	7.900	26.300	39.000	46.900
	Total Time	1.240	1.540	1.850	2.250	0.030	4.260	7.960	26.360	39.060	46.960
Maximal Cliques	Setup Time	28.850	39.160	40.260	50.560	58.820	52.540	39.490	19.650	19.350	19.250
	Solution Time	0.300	0.000	0.000	0.400	0.000	0.000	0.000	0.000	0.000	0.000
	Total Time	29.150	39.160	40.260	50.960	58.820	52.540	39.490	19.650	19.350	19.250

This means that the solver has to run fewer iterations to generate cuts, branches, and bounds, and thus yields a solution in less time. Figure 21 shows this graphically. When looking at the figure, one can see this quite easily. There are also some important notes that should be made related to each formulation.

The Wedge and Core constraint formulation does not perform well for large separation standards when used to represent the DACLP. The reason for this is that with the exception of the pairwise model, the core & wedge model is, in general, larger than the other models. Those formulations that use a combination of neighborhood and clique constraints generally receive a benefit in that the overall problem representation is small. For example only two constraints need be written for each facility site for the Big-*M* and Yoshimoto and Brodie formulations, and at most three for the Erkut et al. (1996) constraint forms. This results in fairly compact, tight formulations that are easily solved. Given the computational experience of the ACLP and DACLP constraint formulations, it is likely that if one needs to solve for both forms of the ACLP and DACLP the constraint types to use would be that of Erkut et al. (1996). This is because the overall solution time when using both forms is fairly efficient in terms of solution time. With regard to the Maximal Clique constraint representation, it again performs the slowest due to the computation time necessary to develop the maximal clique sets.

Figure 21. Total solution times of the various constraint representations of the Disruptive Anti-Covering Location Problem



However if one were to implement the Maximal Clique model for both the ACLP and DACLP, they would technically only need to solve the series of vertex packing problems once for each distance. Unfortunately, even if one were to do that for a dense point set the total solution time would still exceed the time of all the other approaches, except for the Core and Wedge approach when the separation standard is very large. This brings us to an important point. When the number of points is large, the likelihood that one will be able to solve a problem to optimality in a reasonable amount of time, say in the time it would take to get a cup of coffee, isn't high. Figure 20 and Figure 21 support this conclusion as times even for the modest sized 372 node dataset of Ruggles and Church (1996) show.

4. Computational Experience on Varying CPU Hardware Types

The types and power of computing hardware have greatly expanded in the last decade. Computers are now able to use 64-bit integers and it is nearly standard for central processing units (CPUs) to have multiple processing cores, or threads, built into the chip itself as well as containing at least 8GB of memory as a standard. There are two main CPU manufacturers in the x86 computing market. They are Intel Corporation and Advanced Micro Devices (AMD). Both companies have taken different approaches to sell their processors.

Intel has designed multi-core chips with the ability to run two “virtual” threads on a single physical CPU thread. This technology makes use of instruction sets generated from the two virtual threads that are then handed to the physical thread. The two virtual threads are designed to utilize the single physical hardware thread when there is “downtime” between instructions on either virtual thread. The advantage of this approach is that one is able to achieve maximal performance from a physical thread. The downside is that if two processes are running long-term and are numerically intensive on each thread, it can slow performance. Since Intel chips are designed to use hyper-threading, which generates significant amounts of heat and stress to the CPU core when running at full capacity, their chips are clocked at slightly lower speeds but generally contain a greater number of computational threads. AMD, on the other hand, has focused on developing lower cost processors that use only physical threads that run at higher clock frequencies to achieve a similar level of performance.

Given this increase in computational power, many operations research solvers have greatly expanded their computational ability. They do this by taking advantage of the ability to address more memory and to take advantage of parallel processing techniques provided by the ability to take advantage of a multiple thread CPU. This section seeks to describe the computational experience of running several of the ACLP formulations using three separation standards that stress the Xpress solver. In particular, separation standards of 15, 7.5, and 5 were chosen.

Table 7 lists the computational experience of running the Big-*M*, Yoshimoto & Brodie, Pairwise, Core Cliques, Wedge & Core, and Maximal Cliques formulations of the ACLP on the three computers. The separation standard used in these models is 15km. The computers used have the following CPUs: IO is a laptop PC with an AMD A10 5750M 2.5GHz variable-speed mobile quad-core (4 total threads) CPU; Jupiter is a desktop PC with an AMD A10 6800K 4.1GHz quad-core (4 total threads) CPU; SuperChief is a desktop PC with an Intel i7 2.8GHz quad-core hyper-threaded (8 total threads) CPU.

In this case, Jupiter, the quad-core high CPU clock speed computer beat the other machines in all formulations except the Maximal Cliques representation. This is not surprising as these are fairly small problems for which a faster clocked CPU is likely to iterate through faster, particularly given the computational intensity. SuperChief, the computer with more available threads, is the fastest computer when used to solve the several vertex packing problems required to setup the Maximal Clique formulation. This is primarily due to the fact that SuperChief is able to break the setup problem into several smaller chunks that can then be solved. IO, with its mobile CPU, performed the slowest of all three computers. This isn't surprising as this CPU is designed to sip power and balance computation. Though it is slower, it is competitive with the other CPUs, often trailing by less than two seconds of computation time.

Table 7: Computational experience on three different computers for the various ACLP formulations where *r* = 15

	Machine	IO	Jupiter*	SuperChief	Original	Pre-Solve	
Big M	Setup Time	0.040	0.030	0.029	# Const & Variables	# Const & Variables	Objective:
	Solution Time	7.325	5.490	6.048	372	371	6
	Total Time	7.365	5.520	6.077	372	371	
Yoshimoto and Brodi	Setup Time	0.037	0.035	0.028	# Const & Variables	# Const & Variables	Objective:
	Solution Time	7.338	5.474	6.255	372	371	6
	Total Time	7.375	5.509	6.283	372	371	
Pairwise	Setup Time	0.066	0.056	0.053	# Const & Variables	# Const & Variables	Objective:
	Solution Time	8.213	5.750	6.401	69888	4766	6
	Total Time	8.279	5.806	6.454	372	372	
Core Cliques	Setup Time	0.060	0.045	0.043	# Const & Variables	# Const & Variables	Objective:
	Solution Time	1.102	0.787	0.983	744	568	6
	Total Time	1.162	0.832	1.026	372	371	
Core & Wedge	Setup Time	0.070	0.070	0.050	# Const & Variables	# Const & Variables	Objective:
	Solution Time	1.700	1.400	1.900	2119	827	6
	Total Time	1.770	1.470	1.950	372	368	
Maximal Cliques	Setup Time	98.150	84.020	58.820	# Const & Variables	# Const & Variables	Objective:
	Solution Time	0.700	0.600	0.700	745	745	6
	Total Time	98.850	84.620	59.520	372	372	

Table 8: Computational experience on three different computers for the various ACLP formulations where *r* = 7.5

	Machine	IO	Jupiter*	SuperChief	Original	Pre-Solve	
Big M					# Const & Variables	# Const & Variables	
	Setup Time	0.031	0.021	0.021			Objective:
	Solution Time	3.797	3.006	2.745	372	361	18
	Total Time	3.828	3.027	2.766	372	362	
Yoshimoto and Brodi					# Const & Variables	# Const & Variables	
	Setup Time	0.031	0.023	0.022			Objective:
	Solution Time	3.781	2.954	2.747	372	361	18
	Total Time	3.812	2.977	2.769	372	362	
Pairwise					# Const & Variables	# Const & Variables	
	Setup Time	0.031	0.031	0.032			Objective:
	Solution Time	1.391	1.044	1.153	12332	643	18
	Total Time	1.422	1.075	1.185	372	361	
Core Cliques					# Const & Variables	# Const & Variables	
	Setup Time	0.047	0.043	0.032			Objective:
	Solution Time	4.407	3.446	3.601	744	567	18
	Total Time	4.454	3.489	3.633	372	367	
Core & Wedge					# Const & Variables	# Const & Variables	
	Setup Time	0.050	0.050	0.050			Objective:
	Solution Time	1.300	1.100	1.100	2271	642	18
	Total Time	1.350	1.150	1.150	372	361	
Maximal Cliques					# Const & Variables	# Const & Variables	
	Setup Time	47.790	48.470	19.690			Objective:
	Solution Time	1.900	1.500	1.500	745		18
	Total Time	49.690	49.970	21.190	372		

Table 8 shows similar results to those in Table 7, though a separation distance of 7.5km is used here. Again, where the size of the problem is small, Jupiter typically is the winner. Where problem sizes are larger SuperChief tends to be the winner. The exception, once again, is the Maximal Clique formulation that heavily favors SuperChief in its ability to parallelize the vertex packing problems. Table 9 shows similar results as Table 7 & Table 8, except that the problems involve a separation distance of 5 km.

Table 9: Computational experience on three different computers for the various ACLP formulations where r = 5

	Machine	IO	Jupiter*	SuperChief	Original	Pre-Solve	
Big M	Setup Time	0.016	0.024	0.020	# Const & Variables	# Const & Variables	Objective:
	Solution Time	3.265	2.617	2.700	372	341	32
	Total Time	3.281	2.641	2.720	372	342	
Yoshimoto and Brodi	Setup Time	0.015	0.022	0.018	# Const & Variables	# Const & Variables	Objective:
	Solution Time	3.219	2.594	2.901	372	341	32
	Total Time	3.234	2.616	2.919	372	342	
Pairwise	Setup Time	0.032	0.027	0.028	# Const & Variables	# Const & Variables	Objective:
	Solution Time	1.062	0.791	0.827	12332	401	32
	Total Time	1.094	0.818	0.855	372	341	
Core Cliques	Setup Time	0.046	0.037	0.030	# Const & Variables	# Const & Variables	Objective:
	Solution Time	3.735	2.996	3.383	744	527	32
	Total Time	3.781	3.033	3.413	372	353	
Core & Wedge	Setup Time	0.040	0.030	0.040	# Const & Variables	# Const & Variables	Objective:
	Solution Time	1.800	1.400	1.400	2214	432	32
	Total Time	1.840	1.430	1.440	372	341	
Maximal Cliques	Setup Time	46.690	47.390	19.600	# Const & Variables	# Const & Variables	Objective:
	Solution Time	1.500	1.100	1.200	739	739	32
	Total Time	48.190	48.490	20.800	372	372	

These results suggest that if one is solving small problems, a higher clocked CPU with several cores is likely to be faster than a CPU with hyper-threading capability running at a lower clock speed. However, when problem complexity is great and problems are large and parallelizable, a computer with multiple threads at a lower clock speed is likely to solve a problem much faster. This is especially true for problems such as the Maximal Cliques formulation.

5. Concluding comments about solving the ACLP and DACLP

When working with datasets containing a large number of points with large separation standards, solving to optimality may take significant computational effort and may or may not result in finding an optimal solution. This is especially true for large environmental datasets that are often in raster form. For example, the ACLP model can be used to analyze useful habitat and possible carrying capacity. In one problem involving the analysis of spotted owl habitat, Church (2013) found that the problem was too large to solve for optimal ACLP solutions. So, in conclusion, for smaller discrete sets of points, the Core and Core & Wedge models should be used. If one is unable to solve the ACLP or DACLP when applied to large datasets of discrete points or a raster, they will have to rely on heuristic approaches to generate a solution. This next section discusses a heuristic that has been developed as a part of this dissertation.

B. Heuristic approaches to generating solutions to the ACLP

The previous sections presented the computational experience in solving the ACLP and DACLP to optimality. This computational experience demonstrates that as separation standards increase, the number of facility pairwise site restrictions increase as well as the time required to compute the optimal solution. This is problematic, especially in the context of using datasets that have a large number of point locations. This is especially true if one is using a raster dataset containing a significant number of pixels at a fine resolution to represent an area. This all but prohibits one for determining a solution quickly, let alone an optimal one. In many cases a modeler or analyst simply wishes to build a feasible solution to their problem that is hopefully near optimal. Because of this, several heuristic approaches have been developed to solve the ACLP.

Heuristics have been developed to solve a variety of spatial location problems. For example the greedy heuristic method has been discussed at length in the literature. Chvatal used the greedy approach to solve the Set Covering Location Problem (Chvatal, 1979), upon which Feo and Resende improved (Feo & Resende, 1989) and subsequently expanded their approach with Smith to a greedy randomized adaptive search (Feo et al. 1994). Chaudhry, McCormick, and Moon (1986) used the Greedy approach to solve the original Anti-Covering Location Problem.

In addition to the greedy approach other methods have been developed to solve the Anti-Covering Location Problem. Such examples include: greedy randomized adaptive search procedure or GRASP (Feo et al. 1994), genetic algorithms

(Chaudhry, 2006), La Grangian relaxation (Murray & Church, 1997b; Ribeiro & Lorena, 2008a; 2008b), tabu search (Yamamoto, Camara, & Lorena, 2002), Bee Colony Optimization meta-heuristics (Dimitrijević et al. 2012), and evolutionary algorithms (Wei & Murray, 2014). All of these approaches are useful for trying to determine near optimal solutions for problems applied to a set of discrete points. There are a few heuristic approaches that have been designed to solve the ACLP geared toward environmental problems that use raster datasets.

Church (2013) developed two heuristic approaches for the ACLP used to identify and map core habitat. The first heuristic he developed is called the "Random Maximum Scatter Routine." This routine systematically selects raster cells that may be used to serve as territorial centers. Each of the cells may not have an overlapping territory as represented as a circular separation standard. The routine selects at random a starting cell from a set of potential cell locations that have not already been selected as a territory center. Once a cell has been selected, all of the other cells within the range of the separation standard of the selected cell are removed from the candidate territory center list. Once all nearby cells have been removed, the algorithm checks to see if any remaining candidate territory centers exist. If they do the process repeats until no candidate territory centers are left.

The second heuristic approach Church (2013) developed is more sophisticated. This heuristic is described as the Maximal Packing heuristic in the book chapter, but will subsequently be referred to as Packer. The Packer approach involves first populating a list of sites to keep track of those sites contained in a raster that have been calculated as being feasible territory centers. Once this has been done, a site is selected at random from the list. Once this site has been selected, all sites within the separation standard of the initially selected site are eliminated from the list.

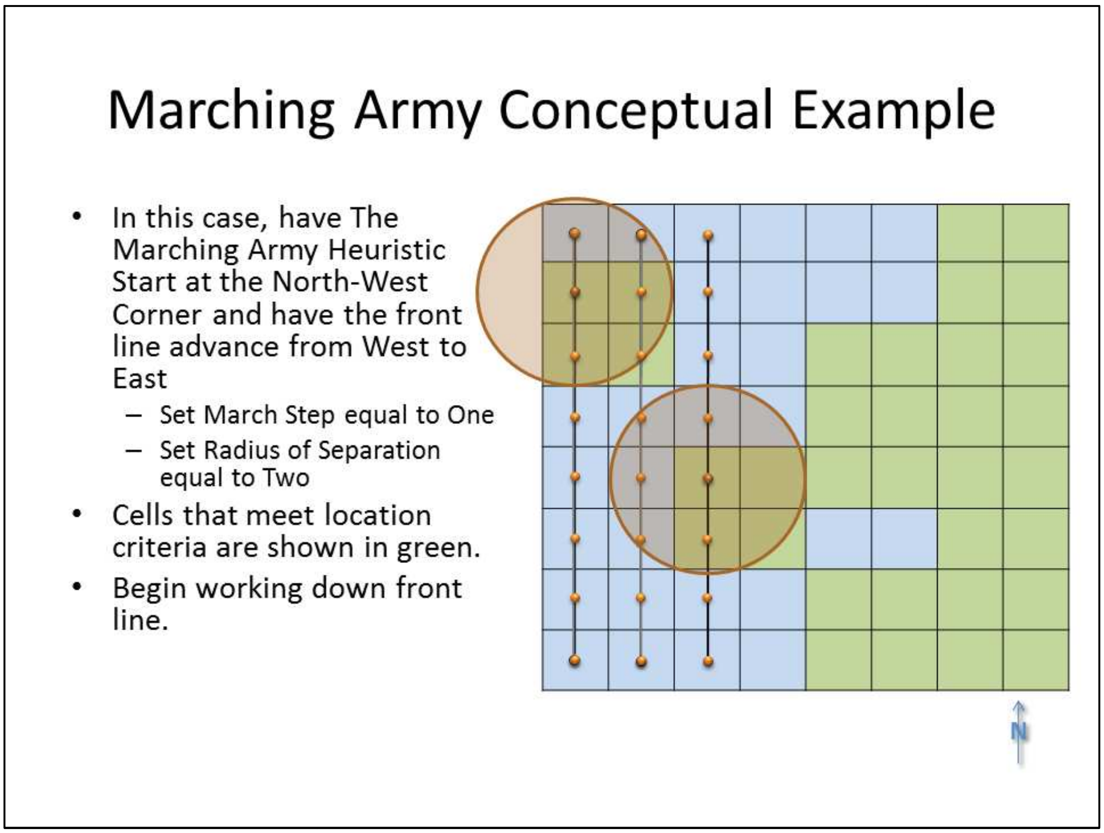
Then the heuristic determines which site on the list is the closest site to the previously chosen site. It then picks the closest site or if there are ties in which site is the closest, it picks at random one of these candidates. This site is selected as a territory center and all of the sites within the separation standard of this selected site are removed from the candidate site pool. The third site that is chosen is the site on the candidate list that is the closest to the first two chosen centers. Each time a candidate is chosen, the list is updated. After the fourth site is chosen, all subsequent sites are chosen based upon their combined distance to the first four sites. This process is repeated until no sites are left in the candidate site pool. In this way a packed solution can be computed relatively quickly for large raster datasets. To date, these two heuristic approaches described in Church (2013) are the only heuristics designed to generate a packed solution using large environmental raster datasets. Both of these approaches, particularly this latter approach, could be used to generate ACLP solutions. Both processes are designed to be repeated a large number of times. However, these two approaches have a few drawbacks and because of this a new heuristic strategy was developed and described in the next sub-section.

1. The Marching Army Heuristic

The Marching Army heuristic is designed to quickly generate a packed solution to an ACLP when applied to a large raster dataset or a set of discrete points. This presentation will focus on its use on a raster or grid of points, where either cells or points are selected one at a time for an ACLP solution. The Marching Army's solution approach can be thought of as an army frontline. The objective is to select as many sites along the frontline for facility location while meeting the minimum separation criteria, and then advance the front. Figure 22 illustrates the heuristic approach over an example raster grid. The simple logic of a Marching Army is provided in Figure 23, and is conceptually related to sweep algorithms, such as that used by Nievergelt and Preparata (1982). To demonstrate the logic as it applies to the sample dataset, consider starting the algorithm at the NW corner of a raster and that the front line runs north to south and advances west to east. The algorithm will check each feasible facility site location along the front working its way down the frontline from north to south. If a site along the frontline is feasible and meets all of the separation requirements associated with previously selected sites, it selects that site for a facility/territory and then moves on. When it reaches and after it considers the last feasible site on the current frontline, the front line is advanced by a march-step, say equal to one cell move eastward, and the process is repeated until all feasible sites along each subsequent frontline advance have been considered and the frontline has reached the other side of the region.

The process can be applied easily and should be repeated using frontlines that run vertically and horizontally (cardinal directions) as well as for non-cardinal directions that could be computed using a linear equation of the form $y = mx + b$. The basic premise is that this heuristic will generate tightly packed arrangements, configurations that meet the type of properties found in optimal solutions. By generating hundreds if not thousands of solutions, the idea is that the best solution generated from this process should be close to if not optimal. Starting locations could potentially be altered to start in the middle of the line or for several starting and ending locations along the frontline, but that would be an extension to this current work. The Marching Army heuristic, as conceptualized here, uses raster data because the problem of interest was defined on a raster.

Figure 22. Conceptual example of how the Marching Army heuristic works



This approach could also be used to generate ACLP solutions on vector datasets in the following three ways. If it was applied to polygon datasets, the heuristic would have to use a point-in-polygon test to determine if it was inside a polygon defined as a feasible center. If a point dataset is to be used, one could first convert the point dataset to a raster such that each point location was uniquely represented as a pixel that the above heuristic could then be applied. Alternatively one could have the heuristic use a buffered line with a very small buffer distance as a front. A series of point in polygon tests on the front could then be used to determine the set of points located on the "frontline". These points could then be selected and packed. These, of course, would be possible extensions in future work.

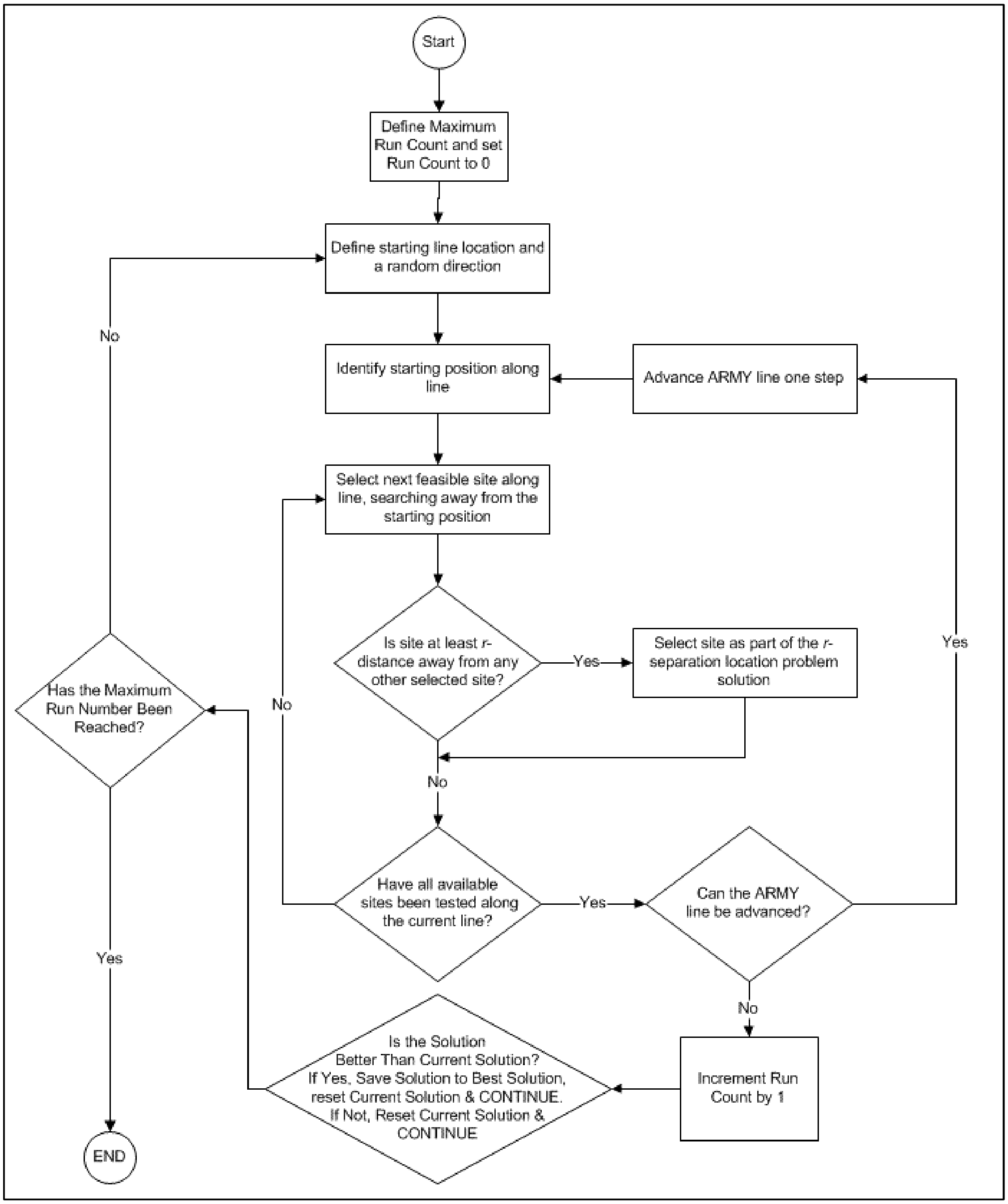
The Marching Army heuristic has the potential to greatly reduce the solution time required to solve an Anti-Covering problem for large raster datasets as compared to using an integer linear programming (ILP) model. In addition, the Marching Army heuristic has the potential to quickly identify optimal or near optimal solutions that current competing heuristics are not necessarily able to find. The heuristic could also be modified to handle the generalized or "almost" *r*-Separation ACLP, as well as the Site Sensitive ACLP proposed in chapter six.

The marching army has been tested on the same dataset in which the two heuristics described in Church (2013) have been applied, the Kings River Protection Area (KRPA) in Sierra National Forest, California. It was developed by Ross Gerrard to identify possible nesting patterns for the California Spotted Owl (*Strix occidentalis occidentalis*), who frequently use locations of dense canopy cover in mature to old-growth forest. This dataset was used to support US Forest Service planning for fire-fuels removal activities and for potential forest disturbances such as fire, so that such impacts to nesting site carrying capacity of the California Spotted Owl population could be inferred. The KRPA raster dataset, representing average canopy about an area of ~2.4 km^2, was generated using a focal mean of ~879 m from US Forest Service vegetation data following the method of Gerrard et al. (2001).

The KRPA dataset consists of ~1,000,000 pixels (an area of ~900 km^2), of which ~75,000 pixels represent potential nesting locations. Average separation distances of spotted owl nest sites is ~1.8km, which means that the focal mean likely captures the required canopy densities for those areas suitable for nesting. Classification of the KRPA canopy cover focal mean raster into high (60-100%), medium (50-59%) and low (0-49%), represent varying levels of habitat suitability. Those locations of high canopy cover are locations deemed suitable for nesting. If one were to attempt to solve this using an ACLP model, it would not be successful as the number of suitable nesting site variables would overwhelm existing solvers. However, one may use a random subset of potential locations in an attempt to derive a solution. Even if one uses a small percentage of the suitable sites, say 7%, the model requires 14.5 hours to derive a solution of 63 sites. Thus, one may be forced to rely on a heuristic approach to obtain solutions to such large datasets in a reasonable amount of time. Furthermore, solving a problem on a sample of sites, rather than the whole dataset, is itself a heuristic.

When one uses a heuristic, it should be run several times with varying starting parameters in order to locate a good solution. The parameters that one should adjust when utilizing the marching army heuristic are the slope of the front line, direction

Figure 23. The Marching Army heuristic logic flowchart

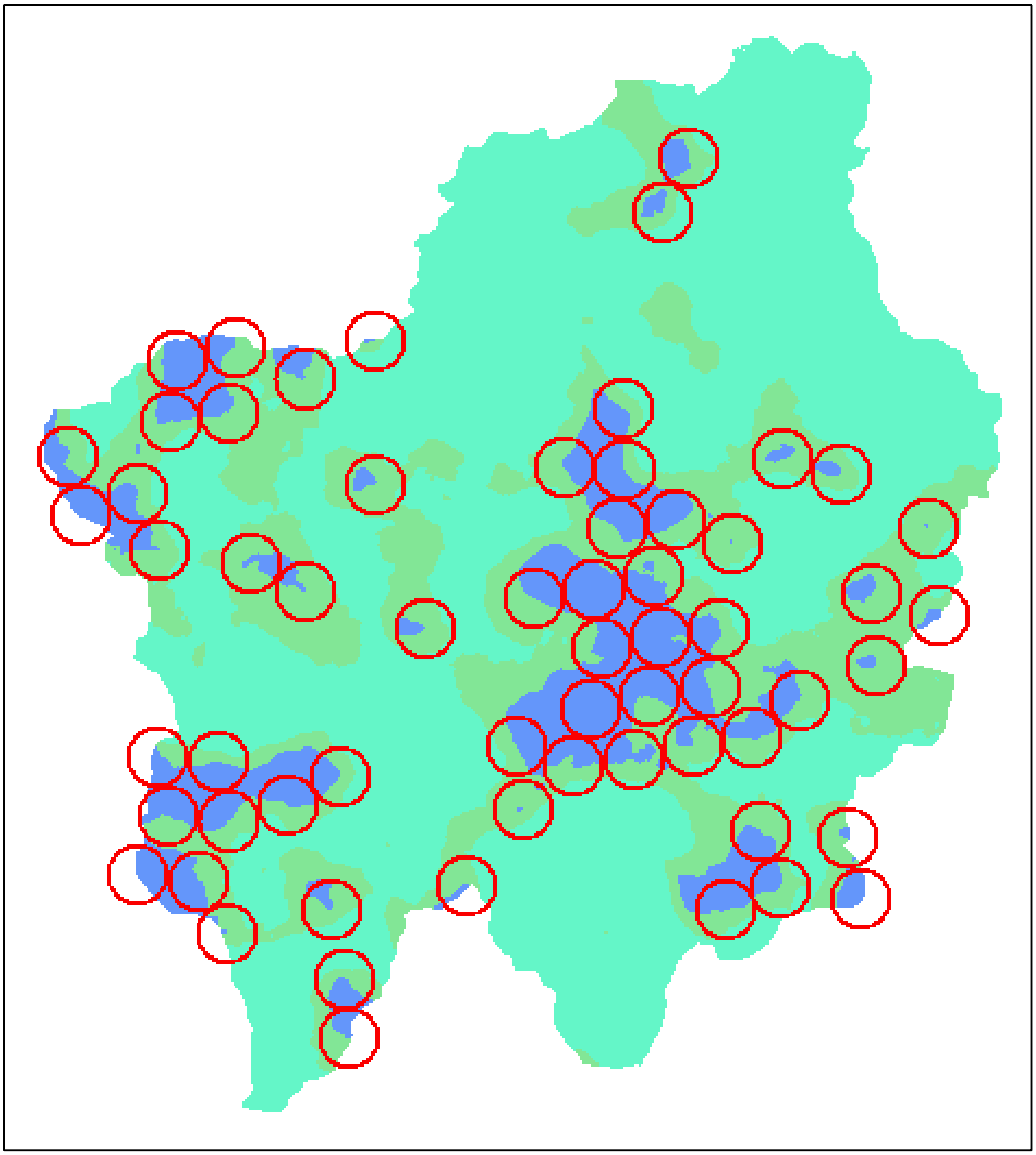


of travel and the starting point on the frontline. In the results presented subsequently, the ends of the frontline were randomly chosen as the starting locations. Any point on the frontline could be randomly selected as a starting point if one desired, though this was not done here. The b-intercept point should be determined by the heuristic so as to start the front-line at a particular point (*e.g.* top or bottom right, or top or bottom left of the study area extent).

Figure 24 shows an example of the solution generated by the Marching Army Heuristic. The figure shows the best packed solution identified by the Marching Army heuristic, consisting of 62 sites. Both the Packer (Church 2013) and Marching Army heuristics found a solution which deployed 62 nesting sites/territories potential facility sites when run 1000 times taking approximately 2.8 hours for Packer and 2.4 hours for the Marching Army heuristics. Figure 25 shows the frequency of objective values obtained by the Packer and Marching Army heuristics. Packer found a configuration of 62 sites 21 times whereas Marching Army found it only 3 times. However, when one examines the histogram presented in Figure 25, it is clear that the Marching Army heuristic on the average finds better configurations. The least packed configuration found by Marching Army is 57 sites as compared to 54 for Packer. In addition, Packer found a configuration of 57 sites on average, while Marching Army found 59. Thus, the spread of solutions is much larger using the Packer heuristic than the Marching Army heuristic and does not do as well on the average. Thus, the Marching Army is a competitive heuristic approach that can identify a solution within 1 site (1.6%) of the best known configuration of 63 sites.

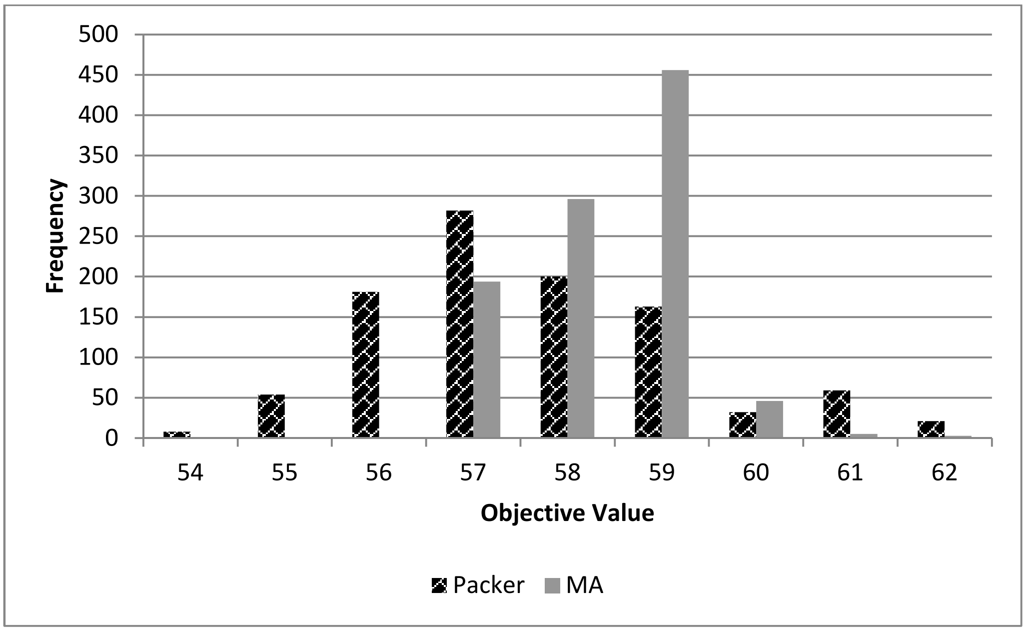
In conclusion, utilizing heuristics to develop solutions to packing problems is necessary for problems where the number of potential location sites is very high. A

Figure 24. This figure shows the Kings River Protection Area in Sierra National Forest, California. The areas represent classifications of the results of a ~879m focal mean average raster of canopy cover. The areas in blue represent dense canopy, a requirement of spotted owl nesting sites. The 62 sites located by the Marching Army heuristic are shown as the red circles.



new heuristic that is fast and produces solutions which are at least near optimality, the Marching Army heuristic, has been proposed and developed. The next section outlines an area of future work that could prove to be useful when developing a heuristic to solve the DACLP.

Figure 25. Histogram showing the frequency of objective values found by the Packer and Marching Army (MA) heuristics.



2. Future Work: Solving the Disruptive Anti-Covering Problem Heuristically

The DACLP is a new location construct. Up to this point, no heuristic has been developed for this problem. The DACLP represents an entirely new way of thinking with respect to the anti-covering location problem. Since the DACLP is geared toward finding the maximally disruptive solution, new approaches should be explored. One approach to solving the DACLP heuristically would be to borrow from heuristic approaches used to solve the Location Set Covering Problem. For example, one could think of trying to "cover" all points within S – 0.001 while keeping all facilities at least S distance apart. Thus, this is an area of future work that should be investigated.

C. Concluding Remarks

This chapter has described the computational experience of solving several existing ACLP formulations and an entirely new formulation, Core & Wedge. The performance of Core and Wedge was tested and compared to existing approaches. Further, it was shown that a neighborhood constraint constant n_i need be no greater than 5. The implementation of this condition subtly improves performance for certain separation standards and problem sizes. In addition to testing this new property, LSCP constraints were tested as possible facets to help improve each formulation. The use of LSCP constraints was to reduce computation times for large problems that have numerous overlapping neighborhoods. This reduction is due to the fact that LSCP constraints, although redundant, can provide valuable cuts to the polytope. A comparison between the varieties of available computing hardware configurations with regard to multi-threaded programs is also given.

This chapter has also described a new heuristic solution approach to the ACLP. This approach works very efficiently and obtains solutions for large raster datasets in less than a second. Furthermore, the solutions it obtains for such large datasets are close to, if not optimal. Future research directed toward development of a heuristic to solve the DACLP has also been proposed.

In conclusion, this chapter has:

- Described the computational experience of:
    - Solving the Core & Wedge formulation for the ACLP and DACLP
    - Solving existing formulations for the ACLP and DACLP.
    - Testing constraints using the updated neighborhood constraint proof derived from the logic of Core & Wedge that shows that n_i is at most 5
    - Testing the efficacy of adding LSCP constraints to a problem to redundant, but strong facets.
    - Testing the performance of hardware configurations of multi-core and threaded CPUs and the use of the Xpress solver
- Introduced a completely new heuristic – Marching Army
- Suggested an approach that may be useful for solving the DACLP heuristically

VI. A Review of What Has Been Covered

This dissertation has focused on facility location problems that involve a separation standard. The dissertation contains: 1) a review of the literature related to dispersion problems; 2) Provides several state-of-the-art advancements in modeling approaches related to the Anti-Covering Location Problem (ACLP); 3) Proposes a new location problem related to the ACLP, the Disruptive Anti-Covering Location Problem (DACLP); 4) Provides the computational experience of solving alternative modeling approaches; 5) Demonstrates a completely new approach to solving the ACLP heuristically; and, 6) provides insights into potential future research directions. The following is a brief synopsis of each chapter and some of the important highlights or points contained within each chapter.

Chapter one sets the stage for this dissertation. It discusses the importance of dispersive strategies in location theory. It further focuses on what dispersion is and provides several examples of dispersive behavior in facility location. One direct example that is readily observed on the earth’s surface is the location of center pivot irrigation systems, as well as several other applications including territorial species carrying capacity modeling (Downs et al. 2008).

Chapter two focuses on the two main conceptualizations of dispersion in location modeling. The first is a distance based approach and the second is a standards based approach. In the distance based approach, the distance between a located facility and all other facilities, the distance between a facility and a set of demands, or both, is maximized. Modeling approaches that use this representation of dispersion are the *p*-

Maxian (Church & Garfinkel, 1978) and *p*-Dispersion (Shier, 1977; Erkut & Neuman, 1991; Lei & Church, 2013) models. The second approach is that of using an explicit standard of separation between facilities. It is this standards approach, implemented in the Anti-Covering Location Problem defined by Moon and Chaudhry (1984) that this dissertation explores.

Chapter three describes the various ways that the Anti-Covering Location Problem can be represented mathematically. These representations include: Big-*M* (Moon & Chaudhry, 1984), refinements to the neighborhood constraints following Yoshimoto and Brodie (1994) and Murray and Church (1995), pairwise formulations (Murray & Church, 1997a), core cliques (Erkut et al. 1996), and maximal cliques (Murray & Church, 1997a). An all-new Core & Wedge model is presented. The Core & Wedge formulation is significant as this proves that an ACLP or DACLP defined in Euclidean space may be represented entirely through the use of at most 7 tight clique constraints. In addition to the Core & Wedge formulation, a proof is provided that shows that a neighborhood constraint used in all of the other formulations excluding the pairwise representation has a maximum of five neighbors that can be located within the separation standard of a given site. This is important as this helps create a tighter neighborhood constraint which is used in virtually all ACLP models. Furthermore, the chapter discusses an additional constraint that may be written which enables efficient cuts to be made to the problem polytope. These are desirable as they generally improve optimization solver performance.

Chapter four presents a new anti-covering location problem, called the Disruptive Anti-Covering Location Problem (DACLP). The DACLP is described and a model formulation is developed. This new problem is particularly important to environmental ecology problems where anti-covering modeling has been used to represent territorial space, location decisions that affect franchise store location, planning policy (*e.g.* regulations related to liquor store location or sex offender residency zones), or any other application for which a maximally packed configuration is not necessarily a realistic outcome.

The DACLP involves finding the minimally “packed” configuration, or the “lower bound” for packing solutions. DACLP configurations thwart denser packing arrangements. An approach is proposed to identify if other solutions exist between the densest packing and the least dense packing configurations. In addition to the DACLP, generalized or “almost” *r*-Separation standards are described and a model is proposed for such a case as well as when there may be site specific separation standards.

Chapter five describes the computational experience related to solving the formulations and concepts described in chapters three and four. Solution times required to setup and solve existing representations of the ACLP and the new Core & Wedge representation are provided. In addition, the constraints implementing the tighter neighborhood constraint are tested as well as the additional LSCP constraint used to provide efficient cuts to the problem polytope. Furthermore, the experience of solving the DACLP is provided. The computational experience related to solving the various formulations show that solving the Erkut et al. (1996) representation is likely to be the most robust when solving the ACLP or DACLP models.

Chapter five also examined the usefulness of multi-core CPUs and hyper-threading. Depending on the type of problem one is solving, a CPU with fewer threads running at a higher clock speed may outperform a hyper-threaded CPU running at a slightly lower clock speed. This is strongly related to the overall size of the problem one is solving and the ability to break up such problems.

A new heuristic approach to solving the ACLP is also proposed. This heuristic was developed to use raster data to compute an ACLP solution and is based upon the concept of a moving frontline, like a marching army. This heuristic can generate a near optimal solution in less than a second for a 30km x 30km raster dataset with a resolution of 30m and separation standard of approximately 1.5km. No other ACLP heuristic is capable of generating a solution to such a large problem this quickly. A future research section related to generating heuristics to solve the DACLP is also provided.

In conclusion, this dissertation has explored the concept of dispersion. It has reviewed modeling approaches using separation standards. It has improved the way existing formulations may be formulated by improving the neighborhood constraint representation. In addition, an entirely new formulation of the ACLP has been defined that consists entirely of very tight clique constraints called Core & Wedge. Furthermore, an entirely new anti-location problem has been defined that describes the problem of finding a minimally packed configuration. This represents a lower bound, or least dense packing configuration that is particularly important when the ACLP is used for planning and policy purposes or when applied to ecological modeling.

This is a crucial point, as the ACLP has been broadly applied to several different types of spatial problems. A model formulation is also given that enables one to explore less dense packing configurations that exist between what may be deployed with the ACLP and the DACLP models. Moreover, a heuristic approach is given that enables large scale ACLP solutions to be obtained when using spatially extensive datasets containing hundreds of thousands of candidate locations. Thus, this dissertation has broad applicability to problems related to dispersion and anti-covering location modeling, furthering the capabilities of using this problem construct in geographical analysis.

References

Berman, O., & Huang, R. (2008). The minimum weighted covering location problem with distance constraints. *Computers and Operations Research, 35*, 356-372.

Berry, B. J., & Garrison, W. L. (1958). Functional Bases of the Central Place Hierarchy. *Economic Geography, 34*, 145-154.

Castillo, I., Kampas, F. J., & Pintér, J. D. (2008). Solving circle packing problems by global optimization: Numerical results and industrial applications. *European Journal of Operational Research, 191*(3), 786-802.

Chandrasekaran, R., & Daughety, A. (1981). Location on tree networks: p-centre and n-dispersion problems. *Mathematics of Operations Research, 6*(1), 50-57.

Chaudhry, S. S. (2006). A genetic algorithm approach to solving the anti-covering location problem. *Expert Systems, 23*(5), 251-257.

Chaudhry, S. S., McCormick, S. T., & Moon, I. D. (1986). Locating Independent Facilities with Maximum Weight: Greedy Heuristics. *OMEGA International Journal of Management Science, 14*(5), 383-389.

Christaller, W. (1933). *Central places in Southern Germany.* (C. W. Baskin, Trans.) Englewood Cliffs, NJ, USA: Prentice-Hall, 1966.

Church, R. L. (2013). Identification and mapping of habitat cores. In F. L. Craighead, & C. L. Convis Jr. (Eds.), *Conservation planning: shaping the future* (1 ed., pp. 219-244). Redlands, California, USA: ESRI Press.

Church, R. L., & Bell, T. L. (1990). Unpacking Central Place Geometry I: Single Level Theoretical k Systems. *Geographical Analysis, 22*(2), 95-115.

Church, R. L., & Cohon, J. L. (1976). *Multiobjective location analysis of regional energy facility siting problems.* Brookhaven National Lab, United States Energy Research and Development Administration. Upton, NY (USA): US Government.

Church, R. L., & Garfinkel, R. S. (1978). Locating an Obnoxious Facility on a Network. *Transportation Science, 12*(2), 107-118.

Chvatal, V. (1979). A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research, 4*(3), 233-235.

Cravo, G. L., Ribeiro, G. M., & Lorena, L. A. (2008). A greedy randomized adaptive search procedure for the point-feature cartographic label placement. *Computers & Geosciences, 34*(4), 373-386.

Current, J. R., & Storbeck, J. E. (1994). A Multiobjective Approach to Design Franchise Outlet Networks. *Journal of the Operational Research Society, 45*(1), 71-81.

Curtin, K. M., & Church, R. L. (2006). A family of location models for multiple-type discrete dispersion. *Geographical Analysis, 38*, 248-270.

Curtin, K. M., & Church, R. L. (2007). Optimal dispersion and central places. *Journal of Geographical Systems, 9*(2), 167-187.

Dimitrijević, B., Teodorović, D., Simić, V., & Šelmić, M. (2012). Bee Colony Optimization Approach to Solving the Anticovering Location Problem. *Journal of Computing in Civil Engineering, 26*, 759-768.

Downs, J. A., & Horner, M. W. (2008). Effects of Point Pattern Shape on Home-Range Estimates. *Journal of Wildlife Management, 72*(8), 1813-1818.

Downs, J. A., Gates, R. J., & Murray, A. T. (2008). Estimating carrying capacity for sandhill cranes using habitat suitability and spatial optimization models. *Ecological Modeling, 214*, 284-292.

Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research, 46*, 48-60.

Erkut, E., & Neuman, S. (1989). Analytical models for locating undesirable facilities. *European Journal of Operational Research, 40*(3), 275-291.

Erkut, E., & Neuman, S. (1991). Comparison of four models for dispersing facilities. *INFOR, 29*, 68-86.

Erkut, E., ReVelle, C., & Ulkusal, Y. (1996). Integer-friendly formulations for the r-separation problem. *European Journal of Operational Research, 92*, 342-351.

Feo, T. A., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters, 8*, 67-71.

Feo, T. A., Resende, M. G., & Smith, S. H. (1994). A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, *42*(5), 860-878.

Gerrard, R., Stine, P., Church, R., & Gilpin, M. (2001). Habitat evaluation using GIS: A case study applied to the San Joaquin Kit Fox. *Landscape and Urban Planning, 52*, 239-255.

Grubesic, T. H., & Murray, A. T. (2008). Sex offender residency and spatial equity. *Applied Spatial Analysis and Policy, 1*(3), 175-192.

Grubesic, T. H., Murray, A. T., & Mack, E. A. (2008). Sex offenders, housing and spatial restriction zones. *GeoJournal, 73*(4), 255-269.

Grubesic, T. H., Murray, A. T., & Mack, E. A. (2011). Sex Offenders, Residence Restrictions, Housing, and Urban Morphology: A Review and Synthesis. *Cityscape, 13*(3), 7-31.

Grubesic, T. H., Murray, A. T., Pridemore, W. A., Tabb, L. P., Liu, Y., & Wei, R. (2012). Alcohol beverage control, privatization and the geographic distribution of alcohol outlets. *BioMed Central, 12*(1015), 1-10.

Hakimi, S. L. (1964). Optimum location of switching centers and absolute centers and medians of a graph. *Operations Research, 12*(3), 450-459.

Hurst, M. E. (1972). *A Geography of Economic Behavior: An Introduction.* Duxbury Press.

Joseph, D., Meidanis, J., & Tiwari, P. (1992). Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In *Algorithm Theory* (Vol. 621, pp. 326-337). Springer.

Kuby, M. J. (1987). Programming Models for Facility Dispersion: the p-Dispersion and Maximum Dispersion Problems. *Geographical Analysis, 19*, 315-329.

Lei, T. L., & Church, R. L. (2013). A Unified Model for Dispersing Facilities. *Geographical Analysis, 45*, 401-418.

Meneghin, B., Kirby, M., & Jones, J. (1988). An algorithm for writing adjacency constraints efficiently in linear programming models. In *The 1988 Symposium on Systems Analysis in Forest Resources* (pp. 46-53). US Forest Service Rocky Mountain Forest and Range Experiment Station General Technical Report RM-161. US Forest Service.

Moon, I. D., & Chaudhry, S. S. (1984). An Analysis of Network Location Problems with Distance Constraints. *Management Science, 30*(3), 290-307.

Murray, A. T., & Church, R. L. (1995). Measuring the efficacy of adjacency constraint structure in forest planning problems. *Canadian Journal of Forest Research, 25*, 1416-1424.

Murray, A. T., & Church, R. L. (1996). Analyzing clique constraints for imposing adjacency restrictions in forest models. *Forest Science, 42*, 166-175.

Murray, A. T., & Church, R. L. (1997a). Facets for node packing. *European Journal of Operational Research, 101*, 598-608.

Murray, A. T., & Church, R. L. (1997b). Solving the anti-covering location problem using LaGrangian relaxation. *Computers and Operations Research, 24*(2), 127-140.

New, L., & Fipps, G. (2000). *Center Pivot Irrigation.* Retrieved July 2014, from Texas A & M Library: http://repository.tamu.edu/bitstream/handle/1969.1/86877/pdf_1181.pdf?sequence=1

Nievergelt, J., & Preparata, F. (1982). Plane-Sweep Algorithms for Intersecting Geometric Figures. (M. D. McIlroy, Ed.) *Communications of the ACM, 25*(10), 739-747.

Nishizeki, T., & Chiba, N. (1988). *Planar Graphs: Theory and Algorithms.* New York, New York, USA: North-Holland.

Prokopyev, O. A., Kong, N., & Martinez-Torres, D. L. (2009). The equitable dispersion problem. *European Journal of Operational Research, 197*, 59-67.

Ribeiro, G. M., & Lorena, L. A. (2008a). Column generation approach for the point-feature cartographic label placement problem. *Journal of Combinatorial Optimization, 15*(2), 147-164.

Ribeiro, G. M., & Lorena, L. A. (2008b). Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers & Operations Research, 35*(7), 2129-2140.

Ruggles, A. J., & Church, R. L. (1996). An Analysis of Late-Horizon Settlement Patterns in the Teotihuacan-Temascalapa Basins: A Location-Allocation and GIS-Based Approach. In M. Aldenderfer, H. D. Maschner, M. F. Goodchild, P. A. Burrough, R. McDonnell, & P. Switzer (Eds.), *Anthropology, Space, and Geographic Information Systems* (pp. 155-174). New York, New York, USA: Oxford University Press.

Shier, D. (1977). A Min-Max Theorem for p-Center Problems on a Tree. *Transportation Science, 11*(3), 243-252.

Stephenson, K. (2005). *Introduction to circle packing: the theory of discrete analytic functions.* New York, NY: Cambridge University Press.

Swain, R. (1971). A decomposition algorithm for a class of facility location problems. Ithaca, New York, USA: Cornell University.

Thompson, E. F., Halterman, B. G., Lyon, T. J., & Miller, R. L. (1973). Integrating timber and wildlife management planning. *The Forestry Chronicle*, *49*(6), 247-250.

Toregas, C., & ReVelle, C. (1972). Optimal location under time or distance constraints. *Papers in Regional Science*, *28*(1), 133-144.

Weber, A. (1909). *Über den Standort der Industrien* (1929 ed.). (C. J. Friedrich, Trans.) Tübingen, Germany: Chicago University Press.

Wei, R., & Murray, A. T. (2012). An integrated approach for addressing geographic uncertainty in spatial optimization. *International Journal of Geographical Information Science*, *26*(7), 1231-1249.

Wei, R., & Murray, A. T. (2014). A multi-objective evolutionary algorithm for facility dispersion under conditions of spatial uncertainty. *Journal of the Operational Research Society, 65*, 1133-1142.

Williams, J. C. (2008). Optimal reserve site selection with distance requirements. *Computers & Operations Research*, *35*(2), 488-498.

Wong, W., & Leung, S. Y. (2009). A Hybrid Planning Process for Improving Fabric Utilization. *Textile Research Journal, 79*(18), 1680-1695.

Yamamoto, M., Camara, G., & Lorena, L. A. (2002). Tabu Search Heuristic for Point-Feature Cartographic Label Placement. *GeoInformatica, 6*(1), 77-90.

Yoshimoto, A., & Brodie, J. D. (1994). Comparative analysis of algorithms to generate adjacency constraints. *Canadian Journal of Forest Research, 24*, 1277-1288.