MAXIMIZING THE COST OF SHORTEST PATHS BETWEEN FACILITIES THROUGH OPTIMAL PRODUCT CATEGORY LOCATIONS

by

Thomas Gertin

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Geoinformatics and Geospatial Intelligence

Committee:

______________	Dr. Kevin M. Curtin, Thesis Director
______________	Dr. Tony Stefanidis, Committee Member
______________	Dr. Matt Rice, Committee Member
______________	Dr. Peggy Agouris, Department Chairperson
______________	Dr. Timothy L. Born, Associate Dean for Student and Academic Affairs, College of Science.
______________	Dr. Vikas Chandhoke, Dean, College of Science
Date: ______________	Fall Semester 2012 George Mason University Fairfax, VA

Maximizing the Cost of Shortest Paths between Facilities through Optimal Product Category Locations

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

by

Thomas Gertin
Bachelor of Arts
Virginia Polytechnic Institute and State University, 2007

Director: Kevin M. Curtin, Associate Professor
Department of Geoinformatics and Geospatial Intelligence

Fall Semester 2012
George Mason University
Fairfax, VA

DEDICATION

This is dedicated to my loving parents Ted and Mary, and my wonderful sister Esther.

ACKNOWLEDGEMENTS

I would like to thank the many friends, relatives, and supporters who have made this happen. Drs. Curtin, Stefanidis, and Rice were of invaluable help. Dr. Curtin's expertise in location science was especially helpful in assisting me with some of the difficult problems I faced. In addition, Dr. Curtin was helpful in guiding me throughout the whole Thesis process, which was unfamiliar territory for me. A special thanks goes out to the Department of Geoinformatics and Geospatial Intelligence at George Mason University. Thank you for offering me a convenient and diverse education in one of the most exciting fields out there.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Traveling Salesman Problem .... TSP
Automated Guided Vehicles .... AGV
Cube per Order Index .... COI
Linear Programming .... LP
Mixed Integer Problems .... MIP
Origin Destination Matrix .... OD Matrix

ABSTRACT

MAXIMIZING THE COST OF SHORTEST PATHS BETWEEN FACILITIES THROUGH OPTIMAL PRODUCT CATEGORY LOCATIONS

Thomas Gertin, M.S

George Mason University, 2012

Thesis Director: Dr. Kevin M. Curtin

A location model is introduced that maximizes the shortest Traveling Salesman Problem (TSP) path between facilities. A scenario has been created where retailers that are interested in designing store layouts wish to maximize the amount of time customers spend in their store. Product location data has been collected from a real supermarket and a representative undirected network has been created. An enumeration method will be used to determine the optimal location of three product categories in the supermarket network. The results contain 5 optimal paths, with an improvement of 32 percent over the store's existing shortest path. In addition linear programming and heuristics are used to generate solutions and compare the results to the enumeration method. These same methods can be used in other disciplines to optimize product locations or locate undesirable facilities. Future research can test scenarios with additional constraints or find ways to minimize processing time.

INTRODUCTION

Network analysis is a significant research area if GIScience(Curtin 2007). Network analysis rests firmly on the theoretical foundation of the mathematical sub disciplines of graph theory and topology and can be used to solve spatial problems(Curtin 2007). The new type of network analysis that is being introduced is a network location problem, which involves selecting network locations on an existing network. This type of problem is related the Traveling Salesman Problem (TSP).

The TSP is one of the classic problems in Operations Research and Location Science and is extensively studied(Lawler E.L. et al. 1985). The TSP can be described as finding the shortest route (tour) for a salesman starting from a given city, visiting each of a specified group of cities, and then returning to the original point of departure(Dantzig, Fulkerson, and Johnson 1954). The TSP is a hard combinatorial optimization problem for which to date no optimal algorithm is known(Crowder and Padberg 1980). Often heuristic methods are used to find good solutions to TSPs. The TSP has many applications including areas of production management, vehicle routing, and vehicle scheduling(Crowder and Padberg 1980).

A scenario is being introduced where the cities themselves can be moved. Therefore the TSP solution will change for each different placement. This situation has not been studied extensively in the literature. This paper will focus on where to place products (cities) to maximize the TSP solution. This problem will be referred to as the MAXIMIN TSP problem. The MAXIMIN TSP has potentially many applications including for military, business, government, and sports applications.

The MAXIMIN TSP will be formally described and an objective function will be presented. Different types of solution methods will be described and compared, as they pertain to the problem under consideration. Several avenues have been identified in the literature review process that could reduce the complexity of the problem. Related problems have been found that could lead to an optimal solution. This paper will focus on maximizing the cost of the shortest paths in a retail grocery store application.

Many retail environments try to maximize the amount of time customers spend in their store. In stores like supermarkets, many types of shoppers have a greater chance of filling their shopping carts the more time they spend browsing throughout the store. Marketers have spent a great deal of time and research attempting to determine how customers travel within stores and how they shop in general. Floor layouts strongly influence in-store patterns, shopping patterns, and operational efficiency(Vrechopoulos et al. 2004). Store layout design can satisfy buyers' requirements as well as influence their wants and preferences(Vrechopoulos et al. 2004). As an example, Wal-Mart has experimented by building 85,000 ft^2 stores and 115,000 ft^2 stores that had identical amounts of fixtures and merchandise. Customers ended up spending more time and money in the larger stores(Dunne and Lusch 2007).

Customers tend to not deviate much from the order of the shortest path that connects all of their purchases(Hui, Fader, and Bradlow 2009b). Rearranging the location of product categories would change the order of purchases and paths of the customers. The premise of this research, is that by finding the optimal product category locations within a store layout one can increase the amount of time customers spend in a store. This case study is modeled after a brick and mortar supermarket in order to better apply the findings to the real world. Optimization techniques are used to test different locations for three main product categories. The methods produce the optimal locations that would create the longest shortest path for a shopper to traverse. The network dataset uses time based on geographic distances to model the costs to traverse each edge.

In the next section, the different research that has been conducted in regards to shopping paths and location problems is revised, with a background on the methods used to solve similar problems. This is followed by a description of the general framework, which includes exhaustive enumeration, linear programming, and heuristic techniques. In section 4, the data used is discussed along with how it was produced. Section 5 discusses the empirical results. Section 6 contains a summary of the results and compares the advantages and disadvantages of using different methods. Finally in section 7, additional opportunities for future research are described.

BACKGROUND AND LITERATURE REVIEW

The MAXIMIN TSP is a type of network location problem. Network location problems involve selecting network locations on an existing network such that an objective is optimized(Curtin 2007). These types of problems are highly combinatorially complex, although some recent attempts have been made to integrate GIS and optimal solution software in order to solve these problems optimally(Curtin 2007).

In the standard traveling salesman problem (TSP), the scenario describes a salesman visiting different cities. The type of applications can vary widely, as the TSP has been applied to vehicle routing and computer wiring. The TSP is a problem of combinatorial optimization. Maximizing the cost of shortest paths between facilities through optimal product category locations is also a problem of combinatorial optimization that includes a TSP component. Linear programming is a tool that can be used to solve TSP problems of a certain size efficiently.

Linear programming has been ranked among the most important scientific advances of the mid-20th century and is an integral component of Operations Research(Hillier). Larger problems of combinatorial complexity can be too difficult for linear programming methods to solve. In these cases heuristics can provide a good solution that is not guaranteed to be optimal. A background of linear programming and heuristic procedures is discussed, followed by related problems to maximizing shortest paths.

Linear Programming

Linear Programming is used to solve location science problems optimally, such as the p-dispersion problem, p-median problem, and the TSP. Optimization means finding a best solution among several feasible alternatives(George L. Nemhauser 1966). The representation of a problem in abstract or symbolic form is known as a mathematical model(George L. Nemhauser 1966). Theories of optimization existed long before the development of Calculus. Characterizing optimization problems by mathematical models goes back to the ancient Greeks(Schichl 2004). Nevertheless, the formal development of optimization theory came from calculus. After the invention of calculus, mathematicians worked actively on optimization problems. The theory was developed for mathematical models containing continuous variables and differentiable functions. Although the theory provided solution procedures for problems with several variables, the theory was not adequate to deal computationally with models containing a very large number of variables.

In the 1940s there was a reawakening and change of direction in the study of optimization theory. This renaissance was stimulated by the war effort. Two significant events that occurred around the same time are the work of scientists and mathematicians on military operational problems, and the invention and development of the digital computer. The scientific approach to military problems became the field of study known as operations research. The formulation and solution of mathematical models of optimization is an integral part of operations research. These models of complex logistic, production, and distribution systems are generally characterized by a large number of variables, and are not suited to be solved by calculus.

The modeling and analysis of an operations research problem in general, evolves through several stages(Mokhtar S. Zazaraa, John J. Jarvis, and Hanif D. Sherali 2010). The first phase involves a detailed study of the system, data collection, and the identification of the specific problem that needs to be analyzed(Mokhtar S. Zazaraa, John J. Jarvis, and Hanif D. Sherali 2010). The next step involves representing the problem though a mathematical model. It is important to make sure that the model satisfactorily represents the system being analyzed, while keeping the model mathematically tractable. The next step is to come up with a solution. The proper technique must be selected. After a solution is determined, it can be analyzed, and the problem can possibly be restructured. To apply it to the real world, the solution would be implemented. The solution of problems in this manner is intended as a means of generating alternative solutions to inform the decision-making process. The process should never replace the decision maker.

The Traveling Salesman problem is an integer linear program(Robert J. Vanderbei 2008). This means that some or all of the variables are constrained to be integers. In the TSP, a city is in a certain step of the solution. There are no fractional steps or cities. Generally, integer programming problems are more difficult to solve compared to linear programming problems(Robert J. Vanderbei 2008). However, the simplex algorithm and the capability it provides to efficiently solve a sequence of linear programs is basic to solving integer programs(Chen, Batson, and Dang 2010). Branch and Bound is a search technique that is used to solve integer linear problems once the simplex method finds a fractional optional solution.

The Branch and Bound algorithm involves solving a potentially large number of related linear programming problems in its search for an optimal integer solution(Robert J. Vanderbei 2008). It is possible but unlikely that the solution to the problem has all integer components when ignoring the integer constraints. The simplest strategy would be to round each resulting solution value to its nearest integer value. This is not the best strategy. In fact, the integer solution so obtained might not even be feasible, since we know that the solution to a linear programming problem is at the vertex of the feasible set and so the movement might go outside of the feasible set(Robert J. Vanderbei 2008). The linear programming problem obtained by dropping the integrality constraint is called the LP-relaxation. Since it has fewer constraints, its optimal solution provides an upper bound on the optimal solution to the integer programming problem(Robert J. Vanderbei 2008). A sequence of relaxations can more tightly constrain the solution space and lead to an integer solution.

Heuristics

Heuristic techniques can be used to obtain a good solution or an approximate solution for an integer program or a combinatorial optimization problem(Chen, Batson, and Dang 2010). Heuristics are part of generally recommended solution strategies to develop good, approximate solutions as well as tighter lower bounds(Chen, Batson, and Dang 2010). Heuristics can be used at each node in branch and cut. Heuristics can become the starting point for an algorithm, significantly reducing the number of iterations to converge to the solution(Chen, Batson, and Dang 2010). Local search, Tabu search, and Genetic algorithms are three heuristic approaches that have been successfully applied to Mixed Integer Problems (MIPs)(Chen, Batson, and Dang 2010).

The heuristic algorithms for TSPs can be classified as construction algorithms, improvement algorithms, and hybrid algorithms(Kim, Shim, and Zhang 1998). Construction algorithms are those in which the tour is constructed by including points in the tours, usually one at a time, until a complete tour is developed. In improvement algorithms, a given initial solution is improved, if possible, by transposing two or more points in the initial tour. With regards to improvement algorithms, there exist multiple strategies when dealing with interchanges. Hybrid algorithms use a construction algorithm to obtain an initial solution and then improve it using an improvement algorithm.

Greedy algorithms start with a partial solution and repeatedly extend it until a complete solution is obtained. At each stage the locally optimal choice is made. For a given problem there can be many different greedy algorithms corresponding to different ways to define what form a partial solution takes, different forms for the partial solution, and different ways of doing an extension. The nearest neighbor algorithm is an example of a greedy algorithm for solving the TSP. It is as follows: “At each stage visit an unvisited city nearest to the current city.” The nearest neighbor algorithm is also an example of a construction heuristic.

Random search is an extremely basic improvement heuristic. It only explores the search space by randomly selecting solutions and evaluates their fitness(Sivanandam S.N. and Deepa S.N. 2008). This is a simple strategy, and is rarely used by itself. Interesting qualities of random search are that if the solution obtained is not optimal, it can always be improved by continuing to run the algorithm(Sivanandam S.N. and Deepa S.N. 2008). A random search never gets stuck in a local optimum. Finally theoretically, if the search space is finite, random search is guaranteed to reach the optimal solution.

Local search heuristics for facility location problems are extremely straightforward. The idea is to start with any feasible solution and then to iteratively improve the solution by repeatedly moving to the best “neighboring” feasible solution, where one solution is a neighbor of another if it can be obtained by either adding a facility, deleting a facility, or changing the location of a facility.

Interchange heuristics, also known as hill-climbing or substitution heuristics belong to the family of local search. It is when the choice of a neighbor solution is done by taking the one locally maximizing the criteria. An initial feasible solution is selected, often randomly. Then iterations take place, with candidate sites being swapped with existing sites. If an improvement is found in the objective function, the swap is accepted. Further swaps can occur to further improve the objective function. Interchange heuristics can be trapped in local optima. There can be differences in how swaps occur between different interchange heuristics, and different versions of interchange heuristics will perform differently for different problems(Dr. Kevin M. Curtin 2011). Stochastic Hill Climbing iterates by randomly choosing a solution in the neighborhood of the current solution and retains this new solution only if it improves the objective function(Sivanandam S.N. and Deepa S.N. 2008).

The 2-Opt algorithm is a well-known heuristic for the TSP problem. It is a type of local search improvement heuristic. A 2-Opt exchange consists of eliminating two edges and reconnecting them in a new way to form a new tour. This is done for all pairs, and the length that gives the shortest tour is picked. This procedure is iterated until no more improvements can be made. The Lin-Kernighan heuristic is the most popular TSP heuristic and is a generalization of 2-Opt and 3-Opt.

Tabu search is a strategy for solving combinatorial optimization problems whose applications range from graph theory to general pure and mixed integer programming problems(Glover and others 1989). Tabu search guides the heuristic to continue exploration without becoming confounded by an absence of improving moves, and without falling back into a local optimum from which it previously emerged(Glover and others 1989). Tabu search uses memory structures that describe visited solutions. If a potential solution has been previously visited within a certain time period or has violated a certain rule, it is marked so the algorithm does not visit that location repeatedly.

Genetic algorithms mimic the process of natural evolution. In a genetic algorithm, chromosomes encode candidate solutions to an optimization problem. The algorithm evolves with each generation to a better solution. The search is only guided by the fitness value associated to every individual in the population. This value is used to rank individuals depending on their relative suitability for the problem being solved(Sivanandam S.N. and Deepa S.N. 2008). If the reproduction operators are just producing new random solutions without any concrete links to the one selected from the last generations, the genetic algorithm is just doing nothing else than a random search.

Similar problems

Warehouse Layout Problems

Comparable in scale to grocery stores, warehouse layout problems have been studied extensively. They consist of a variety of problems including storing, architectural design and general layout problems, picking, response time for order processing, minimization of travel distances in the warehouse, routing of pickers or automated guided vehicles (AGV), and personnel and machine scheduling(Vrysagotis and Kontis 2011). A good warehouse layout configuration may significantly reduce the travel distance for order picking and increase efficiency for successful supply chain operation(Sooksaksun and Kachitvichyanukul). Often the building configuration is already decided and known for an existing distribution center. The necessary remaining steps for layout design are determining the location of stock items and order-picking policy within a distribution center(Liu 2004).

Minimization problems of travel distance in warehouses concern calculating the shortest path for the points in the warehouse where ordered products are kept(Vrysagotis and Kontis 2011). Many different types of mathematical solutions have been used to solve warehouse layout problems including linear programming, particle swarm optimization, and genetic algorithms(Vrysagotis and Kontis 2011). Simulation models have also been used to solve warehouse layout problems (Vrysagotis and Kontis 2011).The basic model that is used is to rank items based on their size and popularity, and locate the smallest and most popular items closest to the input/output (I/O) point(Vrysagotis and Kontis 2011). The Cube per order index (COI) value is widely used to rank the items, it is the ratio of the item’s storage space requirement to its popularity(Vrysagotis and Kontis 2011).

There are typically three storage policies in warehouse operations: randomized storage, dedicated storage, or class-based storage(T. N. Larson, March, and Kusiak 1997). In randomized storage, inventory is allocated to a location, based on the available space at the time of storage(T. N. Larson, March, and Kusiak 1997). In dedicated storage, inventory is assigned to a predetermined location based on throughput and storage requirements(T. N. Larson, March, and Kusiak 1997). Class-based storage is a compromise between randomized and dedicated storage(T. N. Larson, March, and Kusiak 1997). The tradeoff between randomized storage and dedicated storage is that dedicated storage usually reduces the material handling cost, but more total storage space is required(T. N. Larson, March, and Kusiak 1997). The type of storage in a grocery store would be dedicated storage, as the objective of grouping products into similar categories is more important than fitting the most items into the store.

When placing the actual items or storage zones in particular areas, a minimization optimization problem is not usually run. The items with lowest COI rank are specified to be located adjacent to the primary aisle. It is understandable why this simple allocation method exists in placing products in warehouse layout problems. First, the networks are fairly straightforward; this makes it easy to geometrically determine which nodes are closest to the I/O point. Secondly, due to the fact that the costs are being minimized, products that have the greatest demand can simply be placed closest to the I/O point. Complex analysis is not needed.

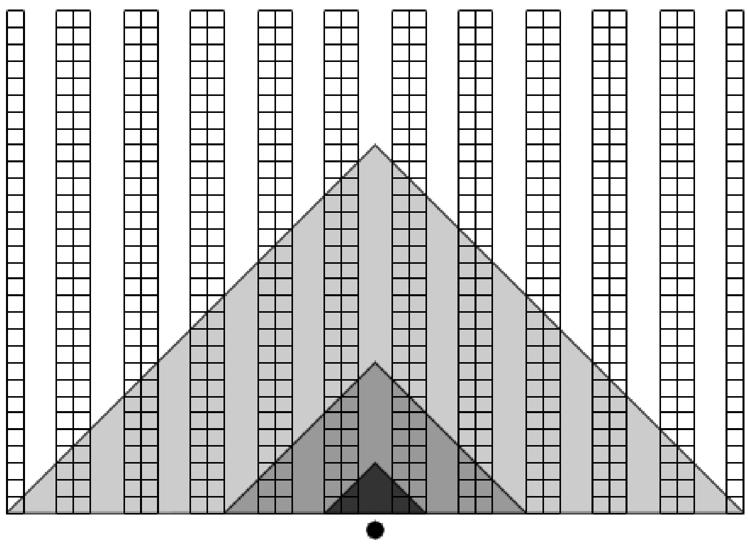


Figure 1. Warehouse Layout Problems. Darker shading indicates more convenient locations for items with better COI value [10].

A need for a better solution becomes more apparent for more complex networks or when the objective is to maximize the cost of the shortest path. A typical grocery store has cross aisles, aisles of different lengths, as well as different sections along the perimeter of the store. This makes it difficult to intuitively see how to spread out the products in the store such that it will create the longest shortest path.

Retail Marketing

Different strategies have been tested in order to increase the amount of time and money customers spend in retail locations. This includes determining the store size, cross-category management, building effective aisle and display management strategies, and store layout design. Retailers have experimented with different sizes of stores including supercenters and hypermarkets. The hypermarket is a retailing format that integrates a supermarket with a department store. It is around one and a half times larger than a typical 140,000 to 160,000 square-foot supercenter. Despite success in Europe and Central and South America, hypermarkets have not been successful in U.S. markets(Dunne and Lusch 2007). However, Wal-Mart's experience with supercenters demonstrated that a larger aisle size permitted greater access to merchandise, which in turn led to greater sales(Dunne and Lusch 2007). Most recently, Wal-mart is planning on opening many Neighborhood Market and Marketside stores. The Neighborhood Market stores run about 42,000 ft^2, and the Marketside stores run about 10,000 ft^2 (Anon.). Retail and store managers must decide how to present merchandise through item allocation and arrangement on store shelves. Through store monitoring, control, and planning they can adjust the workload of each store sector(Bruzzone and Longo 2010). Software and modeling can assist with these tasks. Examples include ShelfLogic (http://www.shelflogic.com); this software generates planograms in order to extract the most from retail displays. 3D visualization can enhance managerial activities such as refurbishment and item arrangement on shelves(Bruzzone and Longo 2010). Path data has been used to capture subjects' eye movement when viewing advertisements(Hui, Fader, and Bradlow 2009a). Cross-category management deals with demand attraction between categories. Retailers can use results of affinity to plan more effective in-store layouts and promotion strategies to increase their customers' cross-buying of products(Bezawada et al. 2009).

Innovative research has been done in studying in-store shopping patterns. In one study, RFID tags were attached to the bottom of shopping carts and emitted a signal every five seconds to receptors placed around the store(J. Larson, Bradlow, and Fader 2005). Multivariate clustering was performed on shoppers' paths to find clusters of paths in the store(J. Larson, Bradlow, and Fader 2005). Using actual real path data, certain myths about shopper travel behavior were dispelled including behavior related to the "racetrack", and traveling up and down aisles. Most shoppers tend to only travel select aisles, and usually travel only partly through an aisle instead of traversing the entire length(J. Larson, Bradlow, and Fader 2005). The perimeter of the store, or "race track", serves as the main thoroughfare and is an area where shoppers spend most of their time(J. Larson, Bradlow, and Fader 2005).

Creating an environment that maximizes the amount of time shoppers spend in the store, can lead to an increase in sales. When focusing marketing strategies towards customers, market segmentation can be useful in grouping customers. The knowledge gained through examining the heterogeneous needs and purchase patterns of customers, enables the retail organization to identify those segments that offer the most promising opportunities(Segal and Giacobbe 1994). Clustering methods have been used by Dr. Herb Sorensen on a large number of shoppers to develop behavioral segmentation(Sorensen 2009). He grouped them into three groups titled Quick, Fill-in, and Stock-up.

Supermarket retailers focus on Stock-up shoppers the most. Stock-up shoppers spend a long-time in the store and cover a lot of area. The fact that there are separate and behaviorally different groups of shoppers highlight the danger of using averages to generalize their characteristics.

Various methods regarding how retailers study and influence customer behavior have been reviewed. Designing successful retail environments involves many different factors, in this paper we focus on store layout and product category location. Store layouts include free flow, grid, loop, and spine. Grid layouts are more commonly used in supermarkets and drugstores(Dunne and Lusch 2007). The grid layout is best used in retail environments in which the majority of customers wish to shop the entire store. The literature supports that it would be valuable to test a store layout after it is implemented to see if any behavioral aspects of shoppers affect the desired outcome.

Location Science

The field of location science deals with the optimal location of facilities, personnel, or services. In 1970, ReVelle published an important paper called "Central Facilities Location." This paper solves a p-median problem and was significant in the development of the field of location science.

Equation 1. p-Median problem

a_i = The population of the i^{th} community
d_{ij} = The shortest distance from community i to community j
x_{ij} = {0 if community i does not assign to community j, 1 if community i does assign to community j}

Objective Function:
Minimize

$$z = \sum_{j=1}^{n} \sum_{i=1}^{n} a_i d_{ij} x_{ij}$$

Constraint 1:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,2,\dots,n$$

Constraint 2:

$$x_{ij} + \sum_{\substack{k=1 \\ k \neq j}}^{n} x_{jk} \leq 1$$

$$i = 1,2,\dots,n$$
$$j = 1,2,\dots,n$$
$$i \neq j$$

Constraint 3:

$$\sum_{k=1}^{n} x_{jk} = 1 \quad j = 1,2,\dots,n$$

Constraint 4:

$$x_{jj} \geq x_{ij}$$
$$i = 1,2,\dots,n$$
$$j = 1,2,\dots,n$$
$$i \geq j$$

Constraint 5:

$$\sum_{i=1}^{n} x_{ii} = m$$

where $m =$ number of central facilities

In the literature review of the paper "Central Facilities Location," ReVelle describes the research that solved similar problems, through heuristic methods. ReVelle's formulation makes use of linear programming to optimally locate central facilities in a road network(ReVelle and Swain 1970). This was at the time period when this problem was receiving greater attention due to advances in mathematical programming and computer-aided computation(ReVelle and Swain 1970). One can take any heuristic solution and tell whether it is optimal with a linear programming solution(ReVelle and Swain 1970).

Most location models deal with minimizing some function of the distance between desirable facilities such as warehouses, service centers, and police stations(Erkut and Neuman 1989). Some examples of obnoxious facilities include nuclear power stations, military installations, and pollution producing industrial plants(Welch and Salhi 1997). These undesirable facilities warrant an analysis that maximizes the shortest distance from customer locations or from other undesirable facilities. In the perspective of the supermarket owner, a model that maximizes the shortest distance between product location categories would be more appropriate. Customers tend not to deviate from the shortest path that connects all of their purchases(Hui, Fader, and Bradlow 2009b). If the shortest path is maximized, the customers will spend more time shopping and have a greater opportunity to fill their carts. When deciding on spatial configuration for modeling, it is important to consider different characteristics such as physical/nonphysical, continuous/discrete, and the presence/degree of constraints(Hui, Fader, and Bradlow 2009a).

Church and Garfinkel dealt with the problem of locating a point on a network so as to maximize the sum of its weighted distances to the nodes(Church and Garfinkel 1978). The literature is filled with instances of problems of locating a facility or facilities close to a given set of points. Finding the location of an obnoxious facility or facilities is not covered as frequently(Church and Garfinkel 1978). Church and Garfinkel state that the extension of the maxian model to the location of multiple facilities is not straightforward. They mention a natural possibility is to let the p-maxian problem be that of locating p points simultaneously far from a given set of nodes and also far from each other.

In the case of non-obnoxious set of location problems, the objective function to be optimized is often of a p-minisum or p-minimax form. In the case of obnoxious facilities the problem is reformulated as a p-maxian/p-maxisum problem or a p-maximin problem(Welch and Salhi 1997).

In the generic single undesirable facility model, the decision maker wishes to locate the new facility such that some measure of the distances between the new facility and the existing facilities is maximized(Erkut and Neuman 1989). Our model will be more similar to multiple facility problems, where there are no existing facilities, and the objective is to maximize some function of distances between new facilities. This problem is also known as the p-dispersion problem. The p-dispersion problem aims to maximize the minimum separation distance between facilities.

Equation 2. p-Dispersion problem

Objective Function:

Maximize D

Constraints:

$$\sum_{i=1}^{n} X_i = p$$

$$D \leq d_{ij} \left(1 + M(1 - X_i) + M\left(1 - X_j\right)\right) \quad \text{for all } i,j \in N \mid i < j$$

$$X_i \in \{0, 1\} \quad \text{for all } i \in N$$

D = smallest separation distance between any pair of open facilities
X_i = {1, if a facility locates at node i}
0, otherwise
n = number of potential facility sites
p = number of facilities to be located
N = set of potential facility sites
M = a very large number
d_{ij} = shortest path distance between node i and node j.

In the p-dispersion problem, there will generally be one pair of facilities that determines the maximin distance(Michael J Kuby 1987). The discrete p-dispersion problem without assuming a specific problem space is NP-complete(Erkut and Neuman 1989). An important difference between the MAXIMIN TSP and the p-dispersion problem is that the aim is not to maximize the *shortest distance* between facilities, instead the goal is to maximize the *shortest path* between all facilities. This objective requires a computationally complex traveling salesman problem to exist within the problem framework.

The maxisum dispersion problem is related to the p-dispersion problem (a maximin problem) in the same way that the p-median problem (a minisum problem) is related to the p-center problem (a minimax problem)(M.J. Kuby 1987). The maxisum problem maximizes the sum of distances (or average of distances) between open facilities(M.J. Kuby 1987). A drawback is that some pairs of facilities could conceivably be placed very near to each other(M.J. Kuby 1987).

Equation 3. Maxisum dispersion problem

Objective Function:

Maximize

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} Z_{ij}d_{ij}$$

Constraints:

$$\sum_{i=1}^{n} X_i = p$$

$Z_{ij} \leq X_i$ for all $i,j|$ j $> i$

$Z_{ij} \leq X_j$ for all $i,j|$ j $> i$

$X_i \in \{0, 1\}$ for all $i \in N$

Z_{ij} = {1, if facilities are located at both i and j}
0, otherwise

p = number of facilities to be located

N = set of potential facility sites

X_i = {1, if a facility locates at node i}
0, otherwise

d_{ij} = shortest path distance between node i and node j.

The p-dispersion and maxisum dispersion problems are two problems that disperse points in some sort of optimal way. It is unknown however if their results will maximize the min TSP path optimally. The p-dispersion and maxisum dispersion problems will be solved optimally and results will be compared to the MAX MIN TSP result solved by enumeration. Solving this problem through enumeration is very hard, due to its combinatorial complexity, and incalculable in many circumstances. If the solution to the p-dispersion or maxisum problem results in a solution that also satisfies the MAX MIN TSP, then it would greatly decrease processing time.

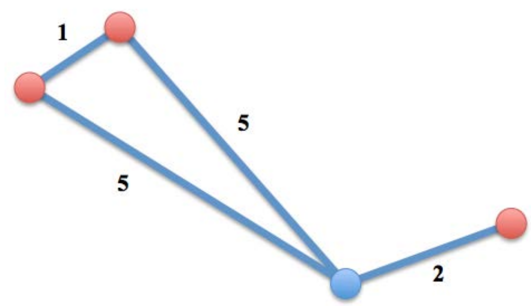
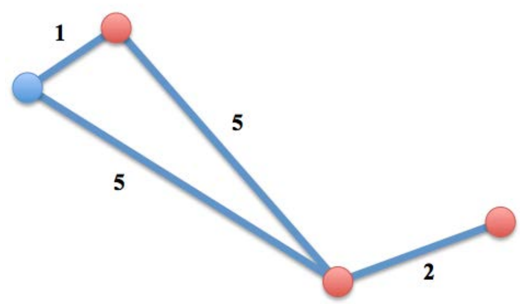
	
Placement A Maximum minimum separation = 1 Sum of distances : 30 =8+8+14 optimal **maxisum** solution TSP solution = 15 * maximizes the TSP solution	Placement B Maximum minimum separation = 2 optimal **p-dispersion** solution Sum of distances : 29 = 12+7+9 TSP solution = 14

Figure 2. Comparison of two facility placements on a simple network

Figure 2 contains two different facility placements on a simple network. Calculating the optimal p-dispersion, maxisum, and TSP solutions for this network can be done by hand. In this network the facility placement that provides the optimal maxisum solution also maximizes the TSP solution. It is unclear if the maxisum always beats the p-dispersion problem when maximizing the TSP solution.

Much work has been done in location theory in the last 60 years, but a small percentage of the research relates to locating undesirable facilities(Erkut and Neuman 1989). An even smaller percentage of research has been related to locating mutually undesirable or several undesirable facilities(Erkut and Neuman 1989). No piece of literature has been found that determines optimal locations for products within a retail setting that maximizes the shortest paths between them.

METHODS

The problem here is defined as finding the longest shortest path between a defined number of facilities in the network. In other words, the optimal solution will be the shortest path with the greatest cost that must be traversed in order to travel to each vertex. It is possible to have multiple optimal solutions. This problem is loosely related to finding the diameter of a graph, a widely used descriptive statistic of graphs. The diameter of a graph finds the longest shortest path between any two graph vertices. To find the diameter of a graph, the shortest path between each pair of vertices must be found. The greatest length of any of these paths is the diameter of the graph. Floyd's algorithm is a graph analysis algorithm for finding shortest paths between all pairs of vertices in a weighted graph(Floyd 1962). A single execution of the algorithm runs with the time complexity of $O(n^3)$, however the algorithm does not include details on the paths themselves. Finding the shortest route between more than two vertices requires the use of a TSP solution and will make the problem NP-complete.

The defined problem has specific criteria and constraints associated with it. The problem is being solved in a network solution space, with network distance measures, using different weights. Multiple facilities are to be located within our network, three to be exact. The problem has only a single objective, to maximize the shortest path between all facilities.

Objective Function

A standard version of the problem requires starting from a given place, visiting subsequent stops, and returning to the starting place. The placement of the given number of subsequent stops can be rearranged on any of the nodes in the network. The optimal solution is one that maximizes the total distance traveled. The objective function (Z) is then used to find the tour with the maximum sum of all costs (distances) from the set of tours in the network that minimize the sum of all costs (distances) of all of the selected elements of each tour:

Equation 4. MAXIMIN TSP problem

n= the number of nodes in the network

r= the number of stops to be visited

y= what is being chosen from the set v

v= the number variations of tours in network:

$$\frac{(n+r-1)!}{r!\,(n-1)!}$$

i, j, k= indices of stops that can take integer values from 1 to n

t= the time period, or step in the route between the stops

x_{ijt}= 1 if the edge of the network from i to j is used in step t of the route, 0 otherwise

d_{ij}= the distance or cost from stop i to stop j

$$\max_{y \in v} \left(\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{t=1}^{n} d_{ij} x_{ijt}\right)$$

The tour is subject to the following constraints:

Since the traveler cannot travel between more than one pair of stops at one time, for all values of t, exactly one arc must be traversed, hence:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijt} = 1 \text{ for all } t$$

For each stop, i, there is just one other stop which is being reached from it, at some time, hence:

$$\sum_{j=1}^{n}\sum_{t=1}^{n} x_{ijt} = 1 \text{ for all } i$$

For all stops, there is some other stop from which it is being reached, at some time, hence:

$$\sum_{i=1}^{n}\sum_{t=1}^{n} x_{ijt} = 1 \text{ for all } j$$

When a stop is reached at time t, it must be left at time $t + 1$, in order to exclude disconnected sub-tours that would otherwise meet all of the above constraints. These sub-tour elimination constraints are formulated as:

$$\sum_{i=1}^{n} x_{ijt} = \sum_{k=1}^{n} x_{jkt+1} \text{ for all } j \text{ and } t$$

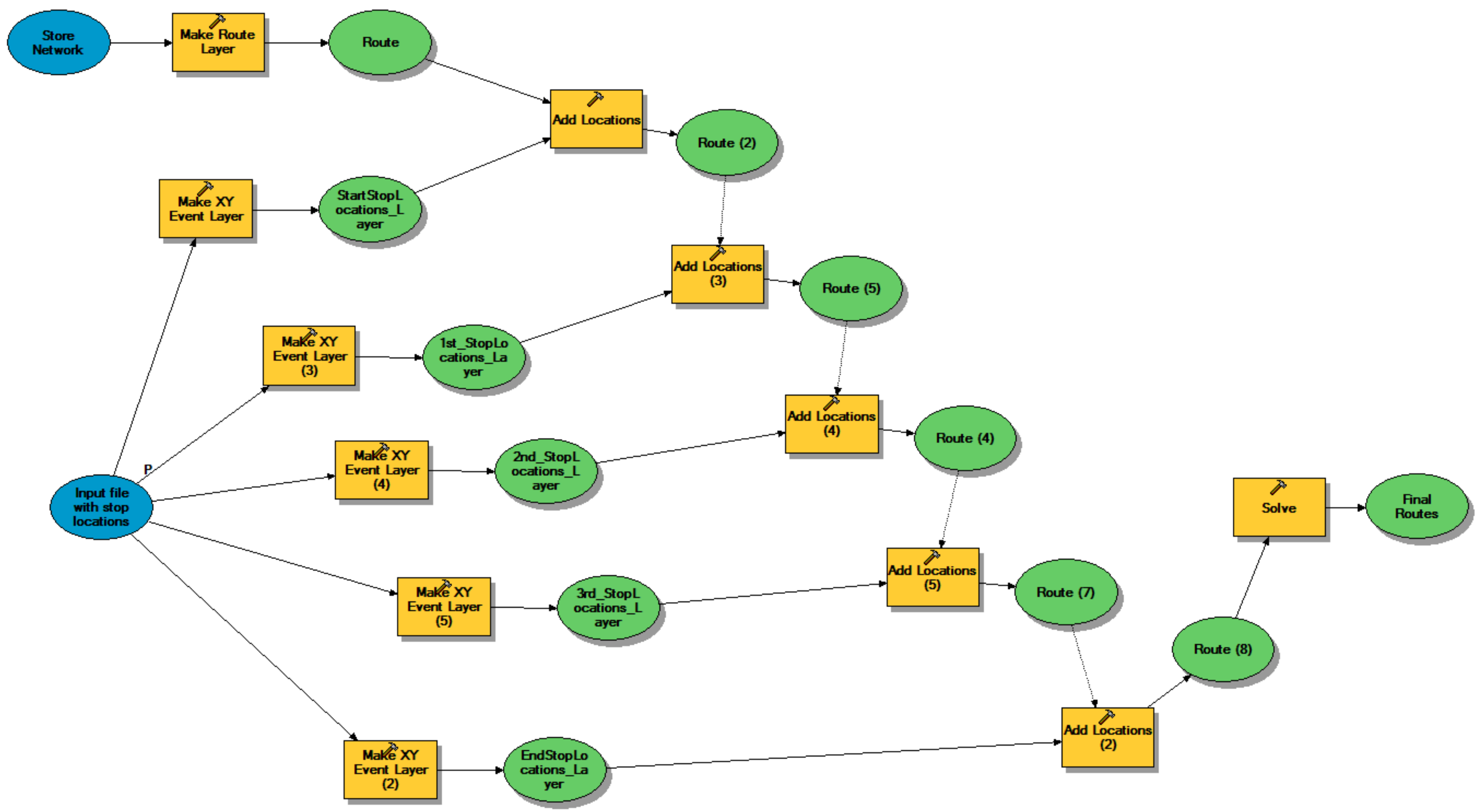
In addition to the above constraints the decision variables must be integers taking only the value 0 or 1:

$x_{ijt} = 0, 1$ for all $i,\ j$, and t

Enumeration

ESRI network analysis tools contain a solver that finds the least-cost route between multiple stops. With 3 stops, the Route solver will need to generate the optimal sequence of visiting the stop locations. This is the Traveling Salesman Problem (TSP). The task of the TSP is to find the shortest possible tour that visits each destination exactly once, given a list of destinations and their pair wise distances. The TSP solver starts by generating an origin-destination cost matrix between all the stops to be sequenced and uses a Tabu search-based algorithm to approximate the best sequence of stops. It has been shown that the ESRI TSP solver achieves optimal results with small datasets(Dr. Kevin M. Curtin 2011).

Figure 3. Model of implementation



To find the optimal solution, Route solver was run for every combination of product category locations for the three product categories throughout the store. In computing the number of combinations in regards to 3 product locations that can be selected from 118 nodes, there are 266,916 different combinations that need to be accounted for. Choosing four product locations would result in 7,673,835 possible combinations, and choosing five product locations would result in 174,963,438 possible combinations.

Equation 5. Combinations without repetition equation

$$C_r^n = \frac{n!}{r!\,(n-r)!}$$

$$C_3^{118} = \frac{118!}{3!\,(118-3)!}$$

For each iteration of Route solver, the start and end nodes are fixed. For 3 products, the output will generate 266,916 shortest paths, the shortest path with the longest total distance the final answer. The nodes that make up this final path will be the locations of the optimal product category locations. A comparison will be made between the optimal solution and where the products are actually placed in the original supermarket layout.

ModelBuilder was used in ArcGIS to generate the computational model. A route layer was made from the store network. Then stop locations were added from five different layers to the route layer. Each layer represented all stops that belonged to a certain stop order. For example, the first layer consisted of 266,916 identical rows of the same node. This was because the first stop represented the supermarket entrance and remained constant for each path combination. The fifth layer also consisted of identical rows representing the supermarket exit. Layers two, three, and four had rows that had a variation of nodes.

Optimization Software

There are many different types of software available to solve linear programming and integer programming problems. These include Excel's solver option, MatLab's optimization toolbox, IBM's Cplex, MPL Modeling System, LINDO, Frontline Systems solver, AMPL and GLPK. The functionality of most of these software packages was explored. Some software had limitations on what types of problems they could handle; others had limitations on the number of constraints they could process.

CPLEX was used to run the p-dispersion and maxisum dispersion problems. CPLEX is a professional solver for linear programming problems (LP), mixed integer linear programming problems (MIP), and quadratic programming problems (QP). The MPS data format was used. The MPS format separates the data into its own separate file. This makes it easier to test the same problem with different datasets, as well as import data from other sources into the problem configuration you are trying to solve.

Heuristics

Local search heuristics start with a given feasible solution and attempt to improve the objective function value by limiting changes in one or a few nodes(Chen, Batson, and Dang 2010). This type of heuristic applies a rule to select an element from a set.

Two types of farthest-neighbor heuristics have been created. They both apply certain rules to select elements from the origin-destination (OD) matrix that contains the shortest paths from all origins to all destinations. The heuristics were written in C#, and the program used the OD matrix as the input in .csv format. A custom program was created named Heuristic Calculator.

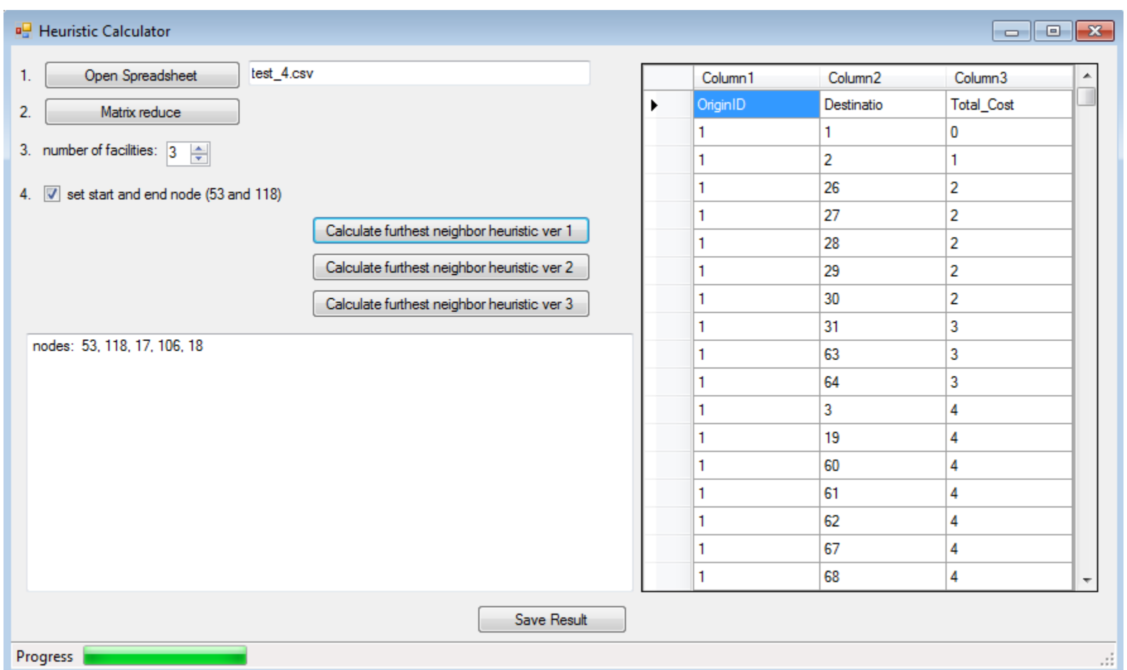


Figure 4. Heuristic Calculator Program

Farthest Neighbor Heuristic

This heuristic orders the distances found in the OD matrix. Depending on X facilities being chosen, the farthest neighbor heuristic finds the largest distances in the OD matrix that satisfies containing x nodes. Each value in the OD matrix is the shortest distance between two nodes. The heuristic has to take into account whether the number of facilities chosen is even or odd. If the number of facilities is even, then the heuristic divides that number by 2 (ex. 4 facilities chosen, 4/2 = 2. Then the heuristic finds the largest X values in the matrix equal to the result (ex. top 2 largest values OD matrix). The heuristic then makes sure that none of the values chosen share a node connection. In the 5-node example, if 5 facilities were chosen, then the top 2 values chosen would be the two connections with a value of 5. These connections do not share node connections, so the 4 nodes that comprise these connections would be selected. These nodes are 1, 2, 3, and 4.

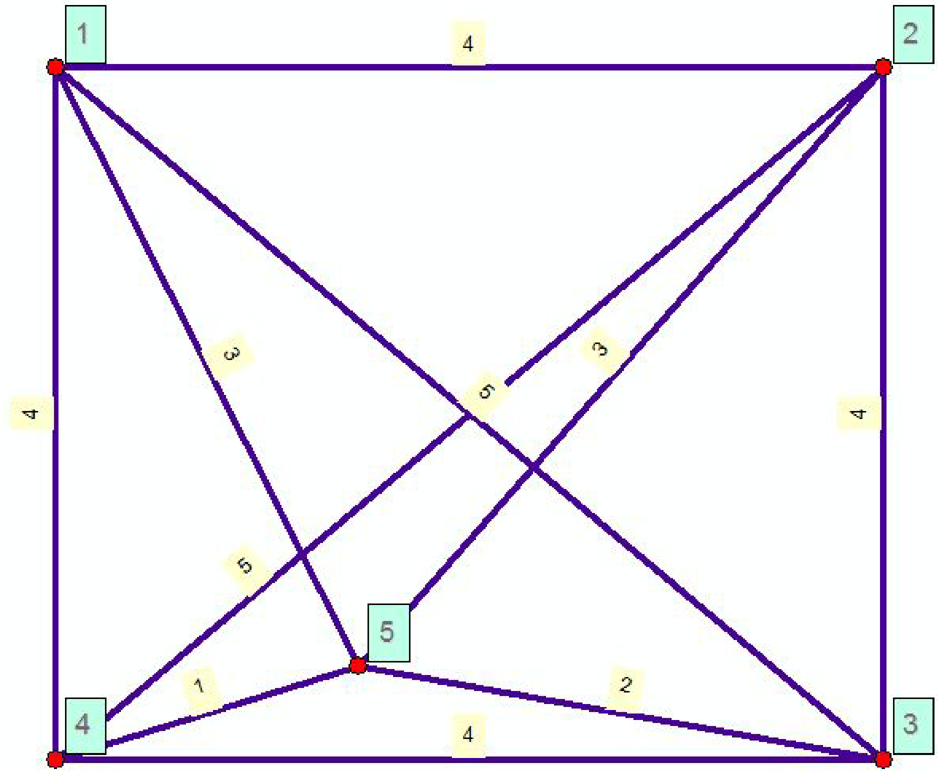


Figure 5. 5-node example for solving the MAX TSP with Heuristics

If an odd number of facilities were chosen, the difference is that one is added to the number of facilities chosen before it is divided by 2 (ex. 5 facilities chosen, (5 + 1)/2 = 3). Also, the heuristic makes sure that there exists one node that shares one connection between two chosen values. In the 4-node example, 3 values would need to be chosen. The two highest values contain a cost of 5. These consist of nodes 1, 2, 3, and 4. The next 4 values could not be chosen because they all share a connection with two nodes that are already chosen. The next value down the list that shares a connection between only one node of a previously selected value is 3. This would ensure node 5 is also included in the solution to the problem. The farthest neighbor heuristic has a significant drawback, the largest values can potentially be close to each other.

The pseudocode for the farthest neighbor heuristic is as follows:

```
USER inputs the number of facilities to be located
USER checks a box to decided whether to set start and end nodes
SET heuristic list = all of the distance values in the OD matrix ordered from greatest to
lowest cost

IF number of facilities chosen to be located is even
    SET X = number of facilities chosen to be located / 2
    IF the checkbox that sets the start and end nodes is checked
        add the start and end nodes to the problem
    END IF
    FOR i = 1 to X
        add the row (node) number from heuristic list(i) to the solution
        add the column (column) number from heuristic list(i) to the solution
        IF solution has any duplicate nodes
            remove duplicate nodes and increment X by 1
        END IF
    ENDFOR
DISPLAY solution
```

END IF

IF number of facilities chosen to be located is odd

SET X = (number of facilities chosen to be located + 1) / 2

IF the checkbox that sets the start and end nodes is checked

add the start and end nodes to the problem

END IF

FOR i = 1 to X

IF there is a need to use a temporary solution list

clear the temporary solution list and repopulate it with the solution list

END IF

add the row (node) number from heuristic list(i) to the temporary solution

add the column (column) number from heuristic list(i) to the temporary solution

sort the temporary solution

IF temporary solution has any duplicate nodes

record if there is one duplicate

END IF

IF i is the last item in X

IF there exists only one node match

add two nodes to solution set, make sure to reset temporary

```
                solution list
            END IF
            IF two nodes are matched
                increment X by one
            END IF
        END IF
    ENDFOR
END IF
```

Farthest Neighbor Pair Heuristic

The farthest neighbor pair heuristic alleviates this problem of having nodes close to each other. The first step of the farthest neighbor pair heuristic is to find the largest value in the OD matrix. This selects the two nodes that are the farthest apart. The next step calculates the sum of the distances between the previous two selected nodes and every other node not selected. The largest sum is chosen to be the location for the next facility. This step is iterated until the required numbers of facilities are chosen.

The pseudocode for the farthest neighbor pair heuristic is as follows:

```
USER inputs the number of facilities to be located
USER checks a box to decided whether to set start and end nodes
```

```
SET heuristic list = all of the distance values in the OD matrix ordered from greatest to
lowest cost
SET X = number of facilities chosen to be located
IF the checkbox that sets the start and end nodes is checked
    add the start and end nodes to the problem, make them A and B
    increment X by two
ELSE
    add the two nodes from the greatest value of the heuristic list as the first two
    nodes in the solution, make them A and B
FOR i = 3 to X
    FOR each node in the network
        sum the distance between the node and node A, and the node and node B
        nodepick = greatest value
        check to see that nodepick isn't a selected node already
    ENDFOR
SET A = B
SET B = nodepick
ENDFOR
```

Farthest Neighbor Sum Heuristic

The farthest neighbor sum heuristic is identical to the farthest neighbor pair heuristic except that in each step the sum of the distances between all of the previously selected nodes and every other node not selected is calculated. The largest sum is chosen to be the location for the next facility. This step is iterated until the required numbers of facilities are chosen. This heuristic was created due to the fact that in the farthest neighbor pair heuristic only taking into account the previous 2 selected nodes can cause the next selected node to be close to an existing node that was selected.

DATA

A local grocery store in Fairfax, Virginia was visited in May 2011 and its store layout was used to create a network graph. All of the grocery store's product categories and locations were documented; in addition several distance measurements were made in order to make a realistic and accurate model. The goal is to find the optimal product category locations using the existing grid layout framework. The shelves, which serve as barriers where customers cannot cross, were taken into account when creating the network. ESRI's ArcGIS has been used for geospatial analysis, creation, storage, and visualization. ESRI's Network Analyst library provides objects for working with network datasets. These objects will allow for network analysis to be performed on undirected networks.

In order to come up with a more realistic scenario, Stock-up shoppers were determined to be the most important group for deciding what product categories should be considered for relocation. Supermarket retailers focus on Stock-up shoppers the most. An annual report from *Progressive Grocer* gave us some insight on who these Stock-up shoppers might be. In 69 percent of households, the female head of household is the primary shopper(Anonymous 2002). The primary shopper tends to make 2 weekly trips to the supermarket and has a major trip time of 54 min(Anonymous 2002). Out of the 18 categories accounted for in the 2002 annual report of *Progressive Grocer*, shoppers spent the most money in fresh meat & seafood, beverages, and produce categories(Anonymous 2002). These three product categories will be used in this analysis, even though the type of product will not affect the results unless a constraint that perishable products can only be located in an aisle with refrigeration is enforced.

There are 222 edges and 118 vertices in the network. The Beta index, a measure of connectivity, is 1.88. Complex networks have a value greater than 1. Another measure of connectivity is the Gamma index, the range of values range from 0 to 1, with 1 representing a completely connected network. The Gamma value of the network is 0.64.

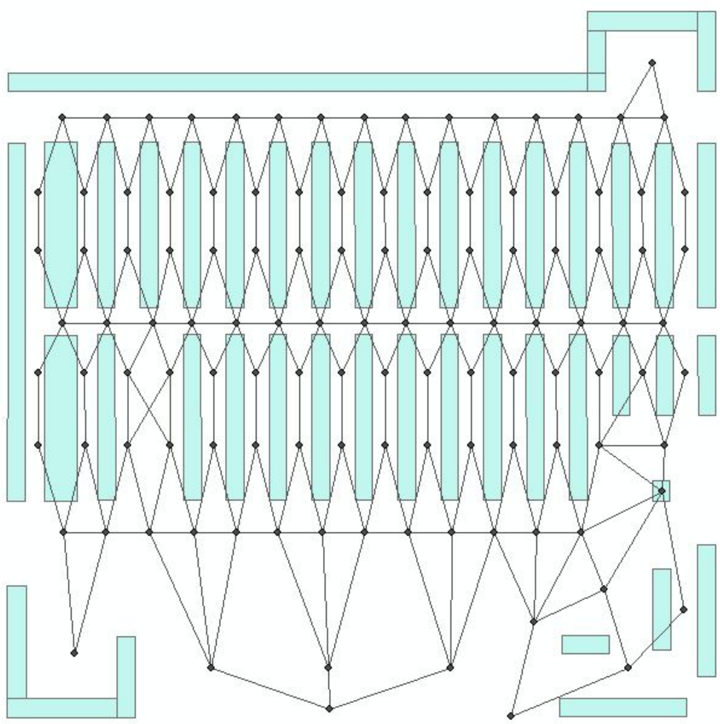


Figure 6. Grocery store represented by a graph of 118 nodes

Our network graph allows us to represent a physical 2-D environment. The aisles located throughout the store create natural barriers to pedestrian movement, yet at the same time limit the amount of possible paths. This lends itself well to being represented by a network graph. This is the same reason why the modeling is discrete, there are finite sections in the store for different product categories that can be represented by nodes. There are no constraints on turns and the graph is planar. While in the physical world a pedestrian's path is limited by the aisles, there are no barriers in the network graph and the appropriate costs of traversal are incorporated in the network graph configuration as well as the costs of the edges. The costs of edges represent 5 seconds or 2.5 seconds for any particular edge.

The shortest path was calculated from each node to each other node. All values were stored in an origin-destination (OD) matrix. Network Analyst in ArcMap is able to generate shortest paths for all origins and destinations. To get the data into the right form for CPLEX, the data was parsed and translated to a matrix form. A custom program called OD Matrix Generator was created, it took an excel file as an input. It parsed each line, which represented a specific origin destination combination and cost, and created an OD matrix in text form as the output.

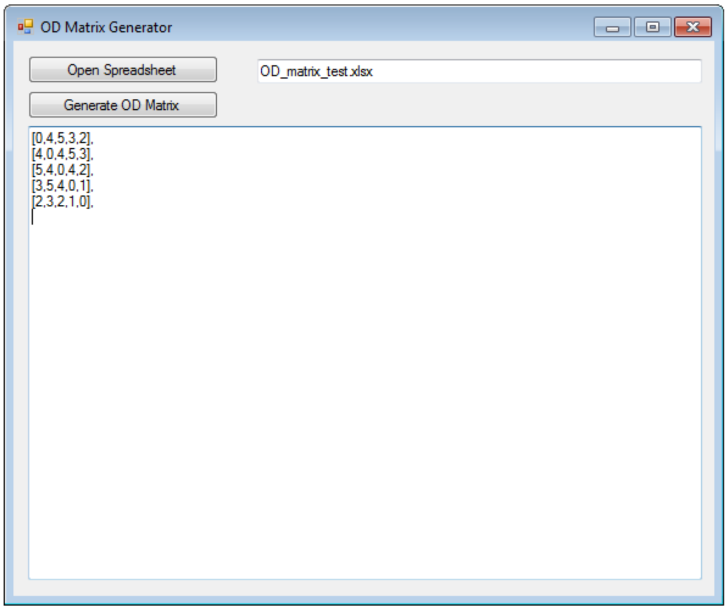


Figure 7. OD Matrix Generator

RESULTS

Enumeration Results

There are five alternate optimal solutions to the problem. All of the optimal shortest paths have a cost of 137.5 seconds. The shortest path of the current grocery store layout was calculated. The shortest path of the current real-world placement is 92.5 seconds, as shown in Figure 5. This results in a 32 percent increase for any one of the five optimal paths. The five different maximum shortest paths are shown in Figures 8 thru 12. Having five different optimal paths will increase the flexibility that store managers have in modifying their store layout. Other criteria not tested for in this experiment can be used to determine which of the remaining five paths the best is. Finally, there is the flexibility of deciding where each product category goes amongst any of the three possible selected nodes.

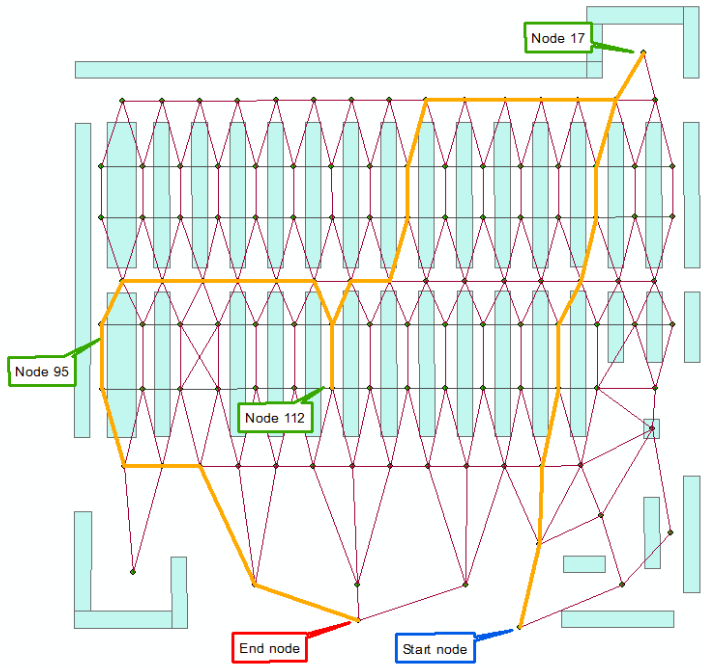


Figure 8. 1 out of the 5 Maximum Shortest Paths

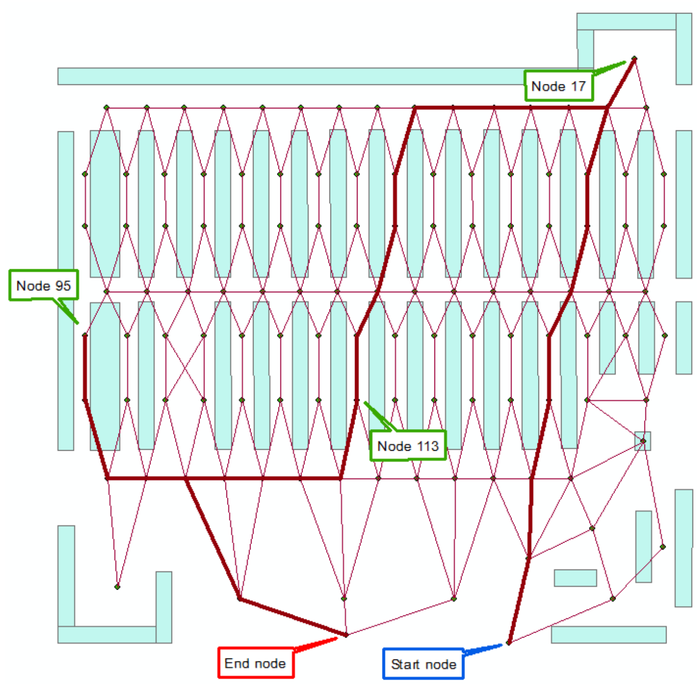


Figure 9. 2 out of the 5 Maximum Shortest Paths

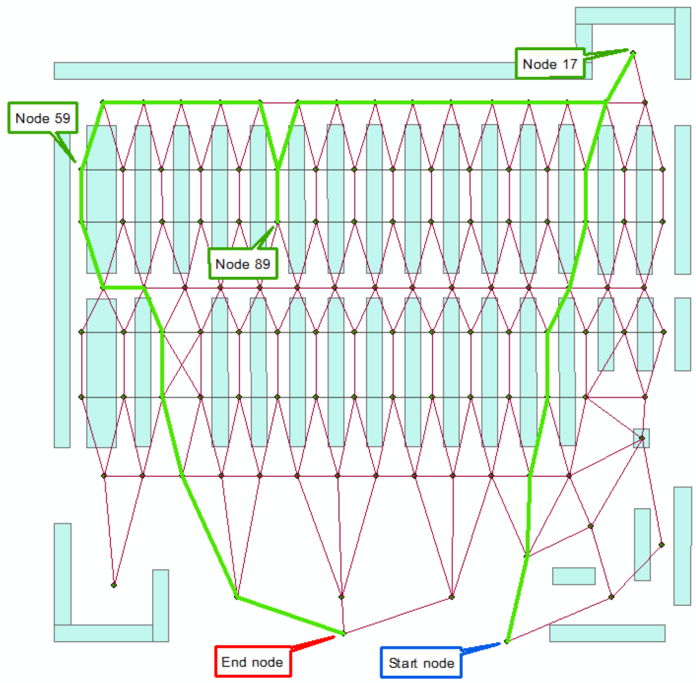


Figure 10. 3 out of the 5 Maximum Shortest Paths

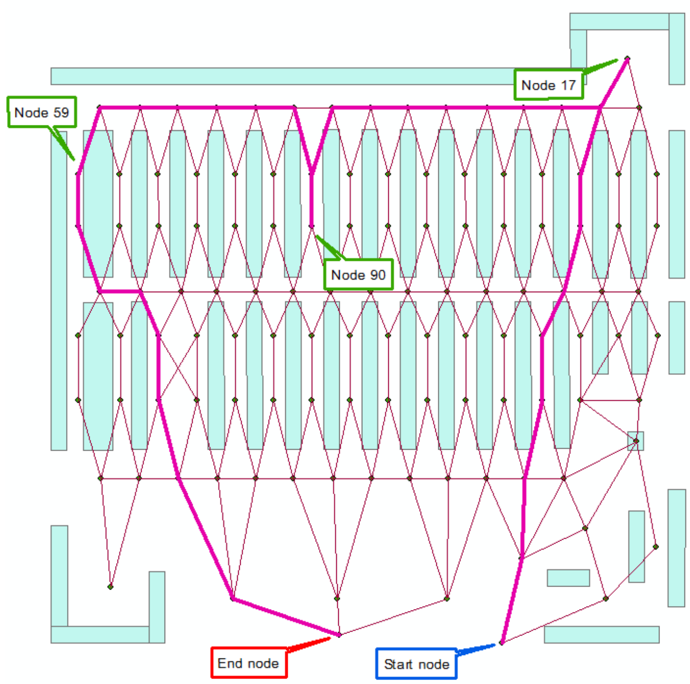


Figure 11. 4 out of the 5 Maximum Shortest Paths

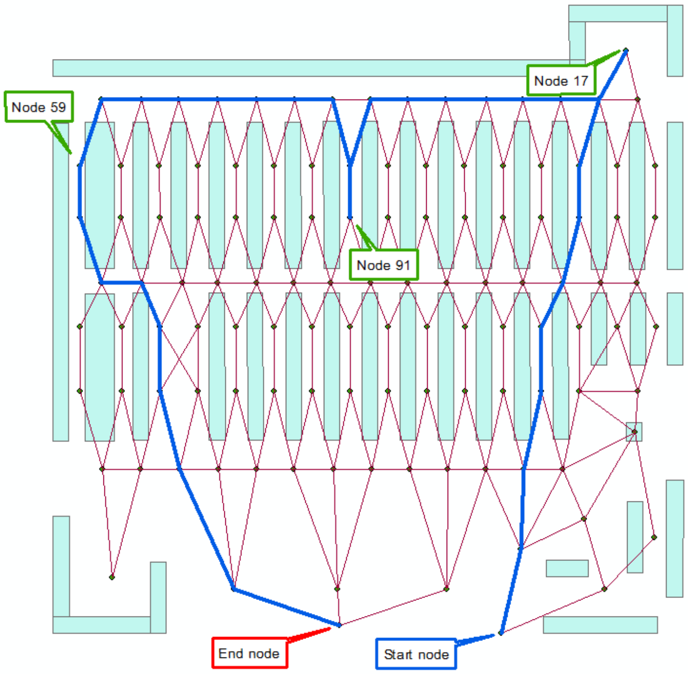


Figure 12. 5 out of the 5 Maximum Shortest Paths

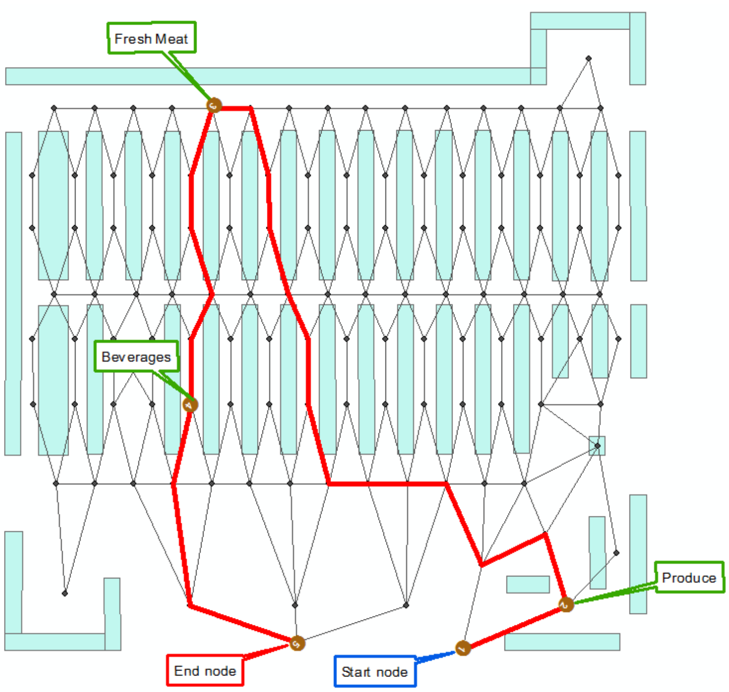


Figure 13. shortest path of actual store layout

There is little variability in the five optimal paths. Node 17 was found in all five paths. In a grouping of three of the five paths, only one of the three facility locations changed. In the remaining two paths, only one of the facility locations was different as well. Node 17, which is the bakery section of the actual store, is at the edge of the graph and requires the customer to backtrack in most cases. All of the optimal paths have the customer cutting across the aisles to reach node 17 as the first destination. It would be interesting to see if customers would actually take this route to get to the first product category or travel through the outside aisle, the traditional 'race track' in many supermarkets.

In several cases the shortest path selection that returns to the checkout nodes requires the customer to switch aisles midway through the store. It would be interesting to note if customers would follow this behavior or stay in the same aisle. This would be a case of travel deviation from the TSP solution, which was found to be a large proportion of trip length in a TSP optimality study (Hui, Fader, and Bradlow 2009b).

The enumeration method took over 7.5 hours to process using ESRI ArcGIS and Microsoft Excel software. The machine was a laptop with a 2.4 GHz i5CPU running on a 64-bit Windows 7 system with 4 GB of ram.

Optimization Results

p-Dispersion Tests

The p-dispersion problem in CPLEX was run with the grocery store dataset for 3 facilities. The Mixed Integer Linear Programming (MILP) objective was 42.5 seconds. The three facilities were located on nodes 17, 53, and 84. The smallest separation distance between any pair of open facilities was between nodes 17 and 54. The TSP path was then found on these three facilities adding the start and end nodes. The cost of the TSP path was 125 seconds.

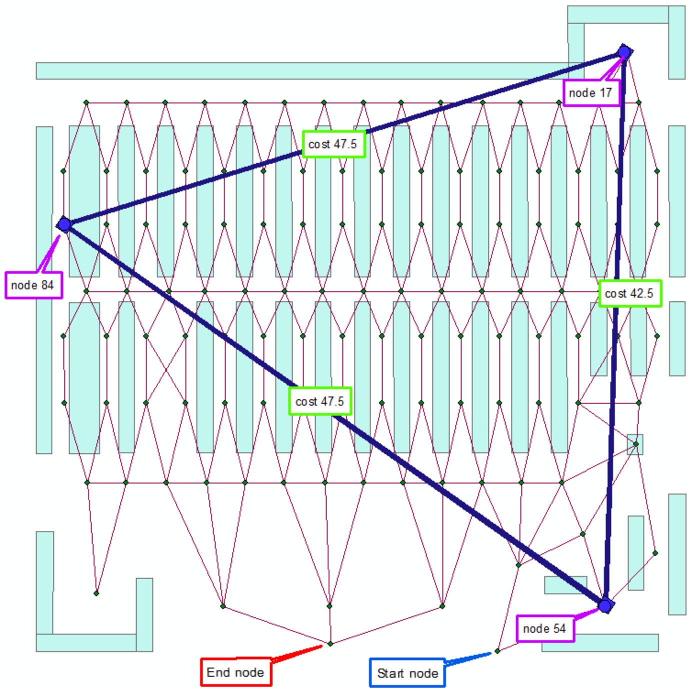


Figure 14. p-dispersion result

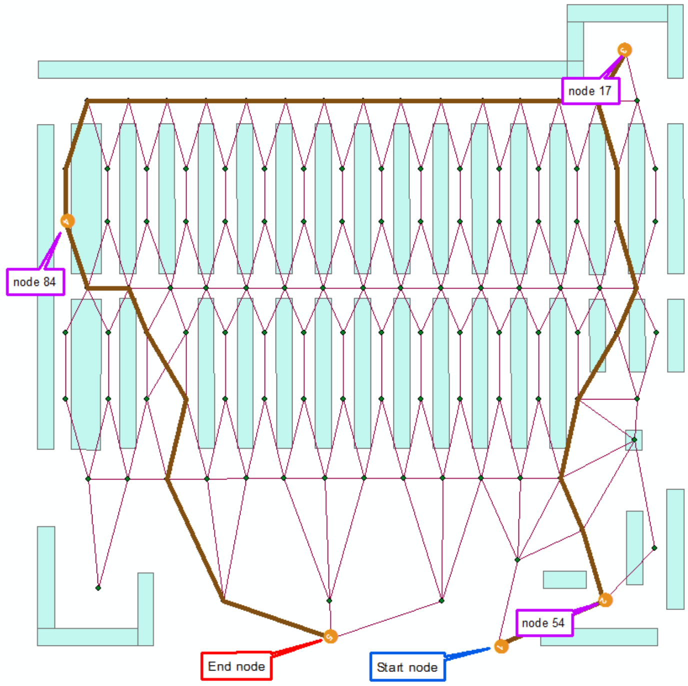


Figure 15. TSP path on p-dispersion result

Next, the p-dispersion problem was run with 5 stops. The start and end stops were fixed. The result had an objective value of 15 seconds. The selected nodes were the start node, end node, 17, 59, and 18. The start and end nodes added an upper limit on the p-dispersion function. The shortest path value between the start and end nodes was already 15 seconds. Therefore the objective value could not be greater than 15 seconds.

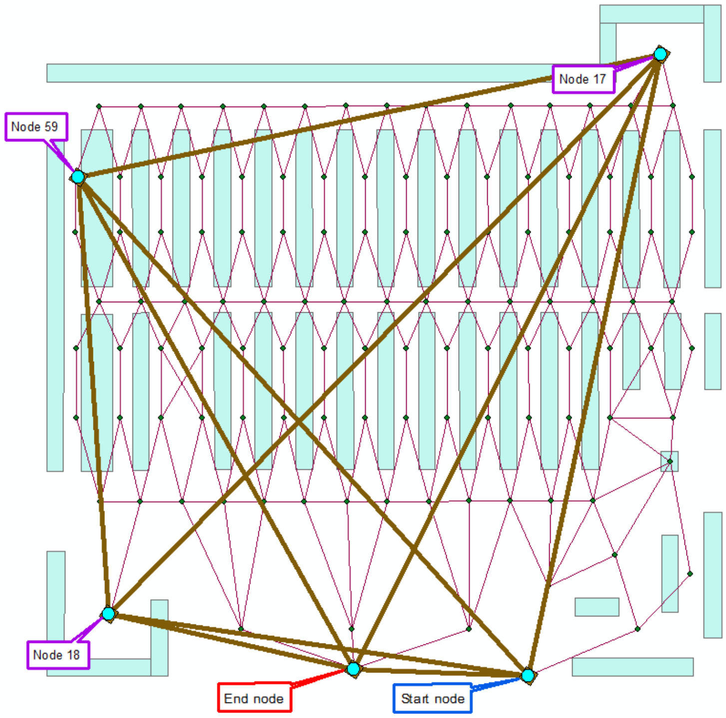


Figure 16. p-dispersion result, 5-stops

Using the nodes from the final result, a TSP solution was run. The total cost was 130 seconds. This was a good result, yet it fell short of 137.5 second optimal path generated using enumeration.

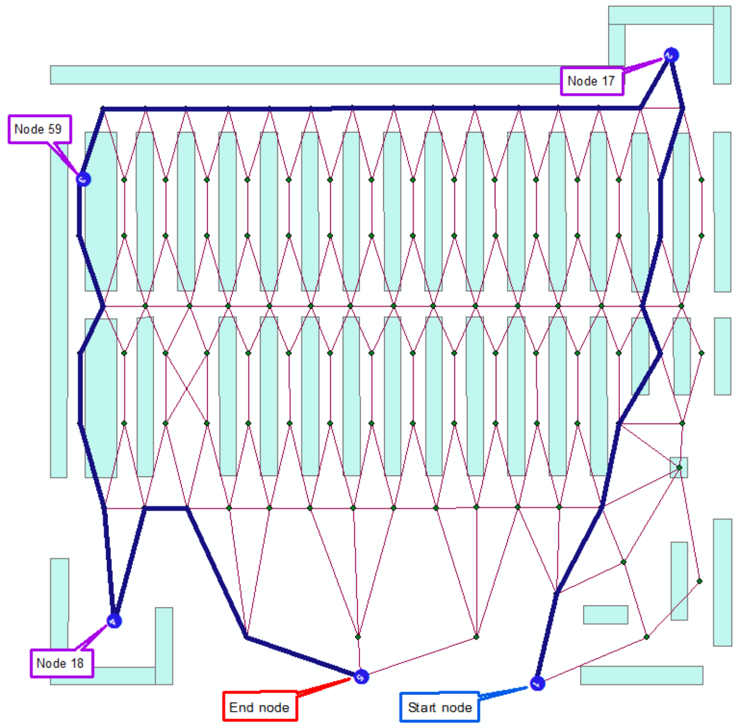


Figure 17. TSP path on p-dispersion result, 5-stops

For the next test, the shortest path between the start and end nodes was artificially lengthened to 60 seconds. The purpose of this was to not let the shortest path distance between the start and end nodes limit the smallest separation distance between any pair of open facilities. The p-dispersion problem was run again, and the final result was 127.5 seconds. Looking at the result, the path is very similar to figure 12, one of the optimal paths generated by enumeration. The difference is that the other path used node 90 instead of 79, 90 was one node further down. This resulted in the 10 second difference between the two TSP paths.

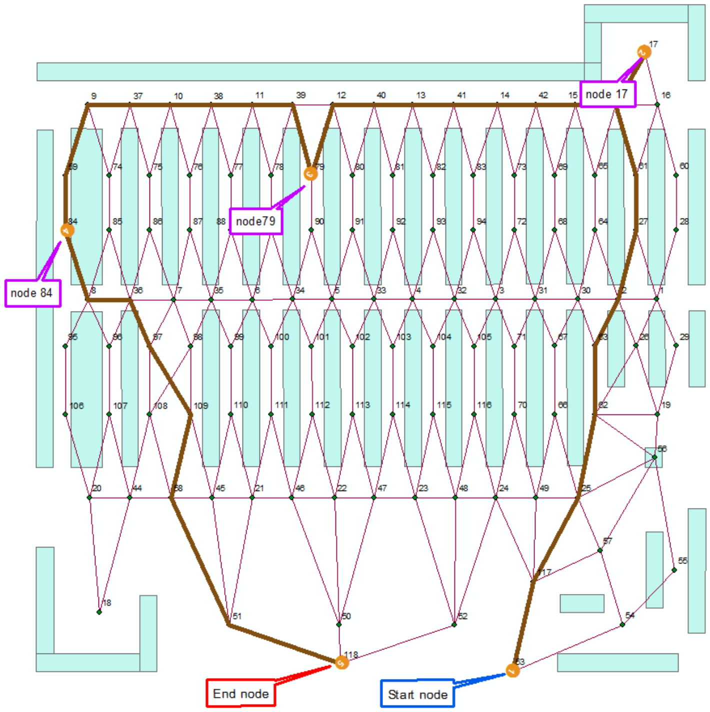


Figure 18. TSP path on 5-stop problem with lengthened shortest path between start and end nodes

Table 1. Processing times for p-dispersion problem

Number of stops	Processing Time
5	3:96 sec
10	6:07 sec
15	5:83 sec
20	32:79 sec

Maxisum Dispersion Tests

The maxisum dispersion problem in CPLEX was run with the grocery store dataset for 3 locating three facilities and setting fixed start and end stops, a total of 5 facilities. The Mixed Integer Linear Programming (MILP) objective was 66887. The selected facilities were the start node, end node, 9, 17, and 19. The cost of the resulting TSP path is 130 seconds. The maxisum dispersion problem was more processing intensive than the p-dispersion problem taking 2 hours 28 minutes and 21 seconds to run with 5 facilities. A 6 facility maxisum dispersion problem could not be completed on CPLEX due to the system running low on memory.

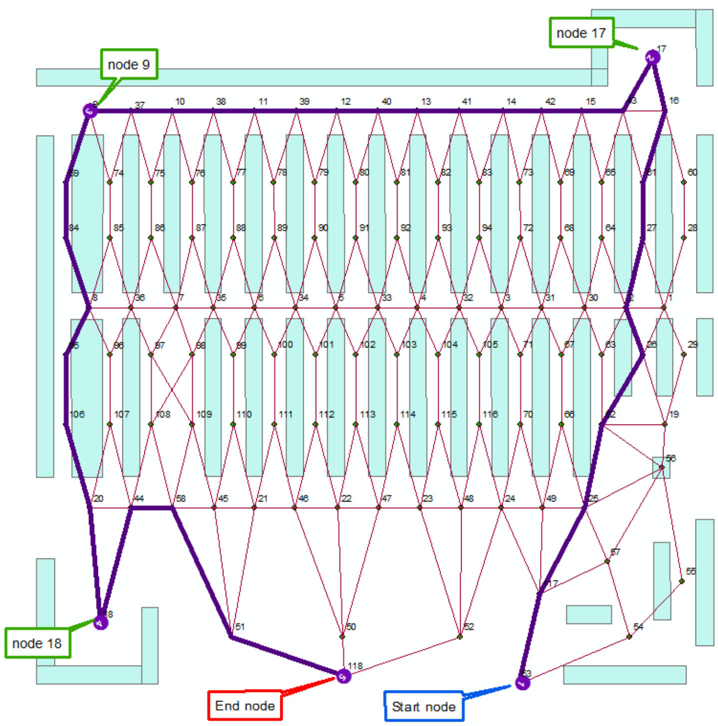


Figure 19. TSP path on maxisum dispersion result, 5-nodes

Heuristic Results

Farthest Neighbor Heuristic

The Farthest Neighbor Heuristic was calculated using the heuristic calculator program. Five facilities were selected with the start and end stops being set to nodes 53 and 118, respectively. The result included nodes 53, 118, 17, 106, and 18 being selected. The cost of the resulting TSP path is 127.5 seconds. The Farthest Neighbor Heuristic took less than 5 sec to run to completion.

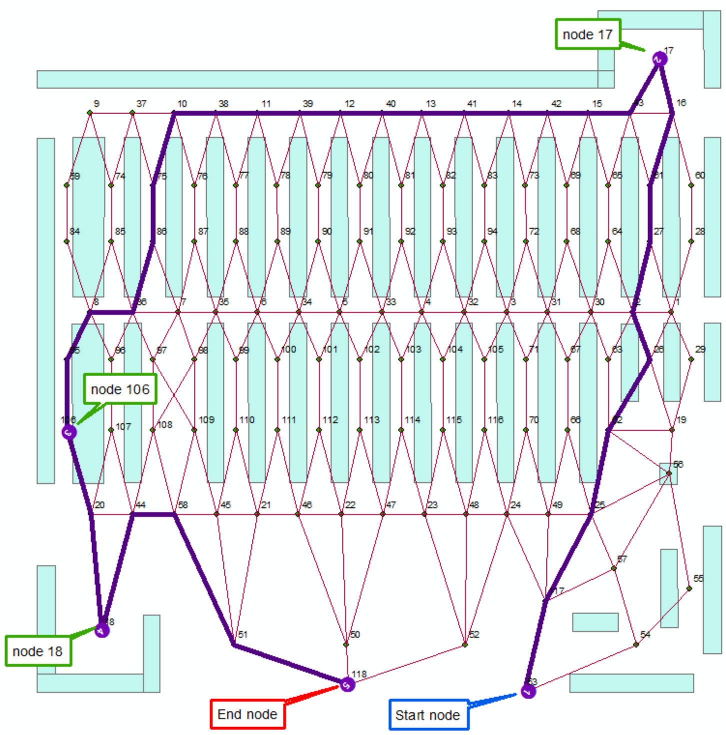


Figure 20. TSP path on Farthest Neighbor Heuristic result, 5-nodes

Farthest Neighbor Pair Heuristic

The Farthest Neighbor Heuristic was calculated using the heuristic calculator program. Five facilities were selected with the start and end nodes being set to nodes 53 and 118, respectively. The final result included nodes 53, 118, 37, 55, and 17 being selected. The cost of the resulting TSP path is 122.5 seconds. The Farthest Neighbor Pair Heuristic took less than 5 sec to run to completion.

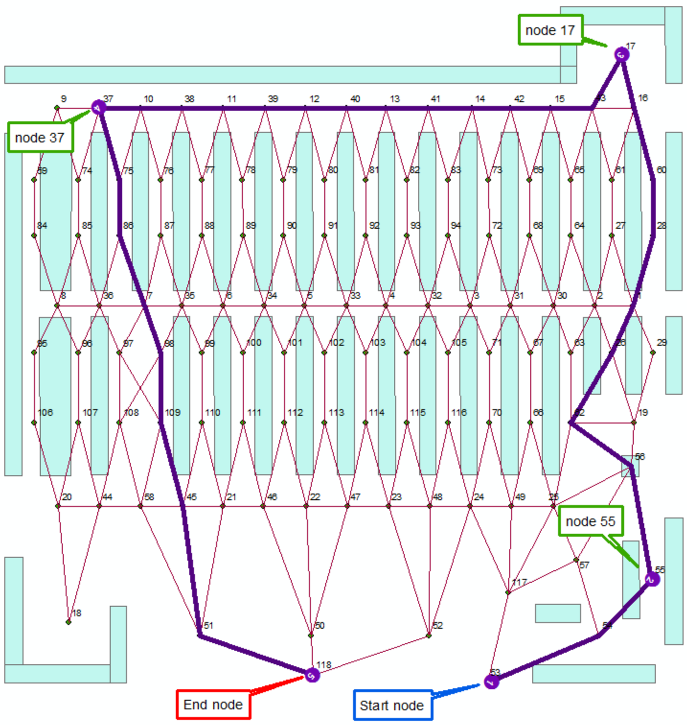


Figure 21. TSP path on Farthest Neighbor Pair Heuristic result, 5-nodes

Farthest Neighbor Sum Heuristic

The Farthest Neighbor Heuristic was calculated using the heuristic calculator program. Five facilities were selected with the start and end nodes being set to nodes 53 and 118, respectively. The final result included nodes 53, 118, 37, 17, and 18 being selected. The cost of the resulting TSP path is 127.5 seconds. The Farthest Neighbor Sum Heuristic took less than 5 sec to run to completion.

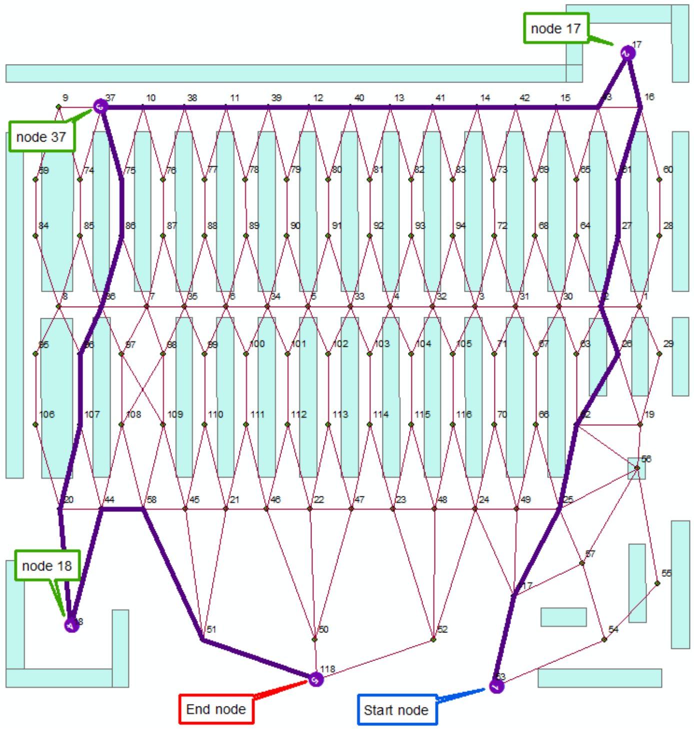


Figure 22. TSP path on Farthest Neighbor Sum Heuristic result, 5-nodes

CONCLUSIONS

In this research, multiple types of analysis were performed regarding combinations of shortest paths between different product category locations. Locations were chosen to maximize the shortest TSP path by enumerating all possible combinations of TSP paths. Undirected graph networks can be used to adequately model and test different product placements in grocery stores. Similar methods could be applied in other types of stores or facilities where optimizing paths would be beneficial.

The enumeration method produced the best result. An important consideration is not to have too large of a network with too many nodes and edges. Adding one more facility to be located would have increased the number of combinations to process from 266,916 to 7,673,835. Using enumeration to solve the MAXIMIN TSP problem can quickly become infeasible with too many facilities to locate.

The p-dispersion problem and maxisum dispersion problems were solved to optimality for their respective problems. The optimal solutions to these problems did not produce a better result to the MAXIMIN TSP problem than the enumeration method. They were close however for the particular network data set used and the number of facilities placed. CPLEX was able to process the p-dispersion problem with up to 30 facilities in under 33 seconds.

The heuristic methods produced competitive results. The farthest neighbor heuristic and the farthest neighbor sum heuristic produced a TSP result of 127.5 seconds. This matched the result of the p-dispersion result, but fell short of the maxisum result. The farthest neighbor pair heuristic fell a bit short with a time of 122.5 seconds. The advantages of the heuristic methods are that they are the fastest to compute. Increasing the number of products to place would increase the processing time by a negligible amount.

Choosing the best method to solve the MAXIMIN TSP would depend on what the requirements are in achieving the optimal solution as well as the processing requirements for implementation. The amount of time needed to generate a solution is dependent on the available hardware. Distributed computing and scalable infrastructure can increase the capability to solve more complex limit. However, there is a limit to what computers can solve when dealing with problems of combinatorial complexity. The MAXIMIN TSP is even more complex than the TSP, which itself is NP-complete. The MAXIMIN TSP requires the TSP to be calculated for each combination of possible product placements in the network.

Achieving the optimal solution might not lead to the desired result in the real world. Human behavior is unpredictable and certain assumptions can fail to take into account the possibility of irrational behavior. The MAXIMIN TSP solution can become a first step in trying various ways to increase profits. Customers have been shown to deviate from the TSP path but not deviate much from the order of the shortest path(Hui, Fader, and Bradlow 2009b). This rational would suggest placing other complementary products with high profits margins where the products were placed when solving the MAXIMIN TSP.

It remains to be seen whether supermarkets would want to implement changing their layout to optimize the longest shortest path. Spreading out commonly purchased items might cause customers to become disgruntled. It is unknown at what limit customers can be inconvenienced that would cause losses in business. Certain physical constraints might prohibit rearranging the store. Perishable items such as meat and dairy need refrigeration; this might limit the possible locations for these product categories.

The MAXIMIN TSP problem can be easily applied problems involving other types of facilities. Amusement parks and convention centers often have an incentive for their visitors to spend more time on their grounds. Networks are used to model a variety of different things including non-physical spaces. Game developers might want players to achieve certain task or get certain items while maximizing the amount of time they spend in a virtual world.

Decisions that involve security issues are another class of problems that be aided with a MAXIMIN TSP solution. Many counties implement security measures for spreading out the necessary materials that are required to create weapons of mass destruction. Valuable information is being stored in critical data centers around the world, including in data centers deployed in theaters. A MAXIMIN TSP solution can aid when deciding where to construct facilities and store materials. Intelligence gathering-activities can focus on documenting activities and patterns of groups of people. This type of data can be represented by networks. Therefore the MAXIMIN TSP can identify crucial locations that can make the adversary incur the greatest cost in acquiring supplies to conduct nefarious activities.

FUTURE RESEARCH

This paper successfully demonstrated the benefits of optimizing the location of product categories. In the process it opened up many possible avenues for future research. There are opportunities to create more complex iterative heuristics and compare the results. TABU search, simulated annealing, and genetic algorithms can be tested.

The next logical step would be to test an optimal solution in an actual supermarket. We could determine first if maximizing the shortest path would actually increase the amount of time shoppers spend in a store. To test this we could attach RFID tags to shopping carts. It would be interesting to compare shoppers' actual paths to the optimal TSP paths. The study of paths with the conjunction of studying different product category locations, may lead to a better understanding of consumer behavior and goals. Point-of-sale data has been matched with the cart movement records to provide data of the items purchased, and to integrate grocery store shopping path and purchase behavior [6].

Testing could be done to see if maximizing the shortest path increases sales, this would not be dependent on having RFID tags. Taking a closer look at a store's purchase history would help us determine market segments better and ultimately select better product categories. Mobile devices equipped with GPS devices are more prevalent, it could be easier to model shoppers' actual paths in an outdoor market. Methods can be tested that add additional constraints; these can include limiting certain node locations for certain products or placing limits on edges. Network costs can take on different values than distance or time, for example it would be interesting to use values that represent products' affinity towards each other. A hypothesis could be proposed that complementary products that have the greatest affinity could be dispersed in order to lure customers to cover a greater distance of the store.

Game theory is a mathematical theory that deals with the general features of competitive and cooperative situations between intelligent decisions makers(Hillier). It would be interesting to apply game theory to the grocery store problem described in this paper. There is a type of competition that arises between the shoppers and the owner of the store, each who have different objectives. It would be possible to formulate this as a two-person, zero-sum game. A sticking point might be that from the shoppers' perspective, they might never be aware of the product dispersion strategy the store owner has at their disposal.

In solving the MAXIMIN TSP problem, the assumption has been that the networks could not be modified. In many real-world scenarios the option exists to modify the network space in order to reach a better solution. In the literature, alternative models have been produced that could reduce the distances workers travel by more than 20 percent compared to the traditional rectangular warehouse design(Gue and Meller 2009). It would be interesting to apply aisle configuration research to my problem. Additional research could test time-sensitivity on optimal networks in different retail outlets. Shoppers' travel time vary across formats(Fox, Montgomery, and Lodish 2004). For example, shoppers at drug stores are more sensitive to travel time than other formats(Fox, Montgomery, and Lodish 2004).

REFERENCES

REFERENCES

Anon. "Wal-Mart's Big Idea For Small Stores - Forbes.com." http://www.forbes.com/2010/09/22/wal-mart-stores-smaller-markets-retailwire-consumer.html.

Anonymous. 2002. "Top of the List." *Progressive Grocer* (April): 27.

Bezawada, Ram, S Balachander, P.K Kannan, and Venkatesh Shankar. 2009. "Cross-Category Effects of Aisle and Display Placements: A Spatial Modeling Approach and Insights." *Journal of Marketing* 73 (3) (May): 99–117. doi:10.1509/jmkg.73.3.99.

Bruzzone, Agostino G., and Francesco Longo. 2010. "An Advanced System for Supporting the Decision Process Within Large-scale Retail Stores." *SIMULATION* 86 (12) (December 1): 742 –762. doi:10.1177/0037549709348801.

Chen, D. S., R. G. Batson, and Y. Dang. 2010. *Applied Integer Programming*. Wiley Online Library.

Church, Richard L., and Robert S. Garfinkel. 1978. "Locating an Obnoxious Facility on a Network." *Transportation Science* 12 (2) (May): 107.

Crowder, Harlan, and Manfred W. Padberg. 1980. "Solving Large-Scale Symmetric Travelling Salesman Problems to Optimality." *Management Science* 26 (5) (May 1): 495 –509. doi:10.1287/mnsc.26.5.495.

Curtin, K. M. 2007. "Network Analysis in Geographic Information Science: Review, Assessment, and Projections." *Cartography and Geographic Information Science* 34 (2): 103–111.

Dantzig, G., R. Fulkerson, and S. Johnson. 1954. "Solution of a Large-scale Traveling-salesman Problem." *Journal of the Operations Research Society of America*: 393–410.

Dunne, Patrick M., and Robert F. Lusch. 2007. *Retailing*. 6th ed. South-Western College Pub.

Erkut, E., and S. Neuman. 1989. "Analytical Models for Locating Undesirable Facilities* 1." *European Journal of Operational Research* 40 (3): 275–291.

Floyd, R. W. 1962. “Algorithm 97: Shortest Path.” *Communications of the ACM* 5 (6): 345.

Fox, E. J., A. L. Montgomery, and L. M. Lodish. 2004. “Consumer Shopping and Spending Across Retail Formats*.” *The Journal of Business* 77 (S2): S25–S60.

George L. Nemhauser. 1966. *Introduction to Dynamic Programming*. John Wiley & Sons, Inc.

Glover, F., and others. 1989. “Tabu Search-part I.” *ORSA Journal on Computing* 1 (3): 190–206.

Gue, K.R., and R.D. Meller. 2009. “Aisle Configurations for Unit-load Warehouses.” *IIE Transactions* 41 (3): 171–182.

Hillier, Frederick. *INTRODUCTION TO OPERATIONS RESEARCH*.

Hui, SK, PS Fader, and ET Bradlow. 2009a. “Path Data in Marketing: An Integrative Framework and Prospectus for Model Building.” *MARKETING SCIENCE* 28 (2) (April): 320–335. doi:10.1287/mksc.1080.0400.

———. 2009b. “The Traveling Salesman Goes Shopping: The Systematic Deviations of Grocery Paths from TSP Optimality.” *MARKETING SCIENCE* 28 (3) (June): 566–572. doi:10.1287/mksc.1080.0402.

Dr. Kevin M. Curtin. 2011. “Location Science” George Mason University.

Kim, B. I., J. I. Shim, and M. Zhang. 1998. “Comparison of TSP Algorithms.” *Project for Models in Facilities Planning and Materials Handling*. https://louisville.edu/speed/faculty/sheragu/Research/Intelligent/tsp.PDF.

Kuby, M.J. 1987. “Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems.” *Geographical Analysis* 19 (4): 315–329.

Kuby, Michael J. 1987. “Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems.” *Geographical Analysis* 19 (4) (October 1): 315–329. doi:10.1111/j.1538-4632.1987.tb00133.x.

Larson, JS, ET Bradlow, and PS Fader. 2005. “An Exploratory Look at Supermarket Shopping Paths.” *INTERNATIONAL JOURNAL OF RESEARCH IN MARKETING* 22 (4) (December): 395–414. doi:10.1016/j.ijresmar.2005.09.005.

Larson, T.N., H. March, and A. Kusiak. 1997. “A Heuristic Approach to Warehouse Layout with Class-based Storage.” *IIE Transactions* 29 (4): 337–348.

Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., and Shmoys D.B. 1985. *The Traveling Salesman Problem*. Great Britain: John Wiley & Sons, Inc.

Liu, C.M. 2004. “Optimal Storage Layout And Order Picking For Warehousing.” *International Journal of Operations Research* 1 (1): 37–46.

Mokhtar S. Zazaraa, John J. Jarvis, and Hanif D. Sherali. 2010. *Linear Programming and Network Flows*. Vol. 4. New Jersey: John Wiley & Sons, Inc.

ReVelle, C. S., and R. W. Swain. 1970. “Central Facilities Location.” *Geographical Analysis* 2 (1): 30–42.

Robert J. Vanderbei. 2008. *Linear Programming Foundations and Extensions*. Vol. 3. Springer.

Schichl, H. 2004. “Models and the History of Modeling.” *APPLIED OPTIMIZATION* 88: 25–36.

Segal, Madhav N., and Ralph W. Giacobbe. 1994. “Market Segmentation and Competitive Analysis for Supermarket Retailing.” *International Journal of Retail & Distribution Management* 22 (1): 38–48. doi:10.1108/09590559410051395.

Sivanandam S.N., and Deepa S.N. 2008. *Introduction to Genetic Algorithms*. Berlin: Springer.

Sooksaksun, N., and V. Kachitvichyanukul. “Particle Swarm Optimization for Warehouse Design Problem.”

Sorensen, Herb. 2009. *Inside the Mind of the Shopper: The Science of Retailing*. 1st ed. Pearson Prentice Hall.

Vrechopoulos, Adam P., Robert M. O’Keefe, Georgios I. Doukidis, and George J. Siomkos. 2004. “Virtual Store Layout: An Experimental Comparison in the Context of Grocery Retail.” *Journal of Retailing* 80 (1) (January): 13–22. doi:16/j.jretai.2004.01.006.

Vrysagotis, V., and P.A. Kontis. 2011. “Warehouse Layout Problems: Types of Problems and Solution Algorithms.” *Journal of Computations & Modelling* 1 (1): 131–152.

Welch, S. B., and S. Salhi. 1997. “The Obnoxious< I> p</i> Facility Network Location Problem with Facility Interaction.” *European Journal of Operational Research* 102 (2): 302–319.

CURRICULUM VITAE

Thomas Gertin graduated from Potomac Falls High School, Potomac Falls, Virginia, in 2003. He received his Bachelor of Arts from Virginia Polytechnic Institute and State University in 2007. He was employed as a Geospatial Analyst for four years. This summer he participated in the student employment program with the National Geospatial-Intelligence Agency. In September, he will be the lead developer for his team that will participate in the DARPA Innovation House Project.