NEW HEURISTICS FOR COMPETITIVE AND HIERARCHICAL FACILITY LOCATION PROBLEM

by

Hale Yege

B.S., Mathematical Engineering, Yıldız Technical University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2008

NEW HEURISTICS FOR COMPETITIVE AND HIERARCHICAL FACILITY LOCATION PROBLEM

APPROVED BY:

Assoc. Prof. Necati Aras ....
(Thesis Supervisor)

Prof. İ. Kuban Altınel ....

Asst. Prof. Temel Öncan ....

DATE OF APPROVAL: 25.01.2008

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Assoc. Prof. Necati Aras for his great support, encouragement, advice and patience throughout this work. Whenever I abandoned myself to despair, with his great motivation, I left his room with a smile in my face. I won't be overstating if I mention that without him I would have been lost.

I am grateful to Prof. Kuban Altınel for his guidance, comments and interest both in the classroom and on the committee.

I would like to thank to Asst. Prof. Temel Öncan for his interest, comments and willingness to serve as my committee member.

I am also grateful to TÜBİTAK for their financial support throughout my master education.

I would like to thank to all members of Boğaziçi University Industrial Engineering Department for their great support. I have to specify that I am proud of being a member of this big family.

I am greatly indebted to my wonderful mother Leyla Yege for her trust, never-ending support and love.

Finally all my grateful and sincere thanks and all my love go to my family and my best friends for their moral support.

ABSTRACT

NEW HEURISTICS FOR COMPETITIVE AND HIERARCHICAL FACILITY LOCATION PROBLEM

A lot of versions of the facility location problem have been studied for a long time. This work considers the combination of two versions Competitive Facility Location and Hierarchical Facility Location Problem.

A company wants to locate hierarchical facilities to a market area where there is a competitor that already located its hierarchical facilities. The objective is to maximize the total net profit which is obtained by subtracting the total cost of constructing a chain from the total captured market share. Hierarchical structure is successively inclusive service hierarchy model and all facilities are assumed to be in two levels. Our main assumption is that a customer splits his/her demand (buying power) among the chains proportional to the attraction level to the closest facility of each chain. This patronizing behavior is a hybrid of probabilistic and deterministic patronizing behavior models.

The contribution of this study is treating the attractiveness of facilities as continuous decision variables. Also the number of each level of facilities to be opened is not predetermined.

A nonlinear mixed integer model is developed for the problem. Firstly the SBB solver within GAMS suite v22.0 is employed. Then a Simulated Annealing Algorithm is developed and within this algorithm two different strategies each having different add-drop criteria are employed as solution procedure. The first strategy consists of add-drop second closest criteria (SASCAD) whereas the second strategy uses random add-drop criteria (SARAD) for neighborhood search. In both strategies Simplex Search

and Fibonacci Search algorithms are employed to find the attractiveness values. Then these strategies are compared and it is seen that SASCAD results better however it is more time consuming. SARAD achieved to catch the same results with the SASCAD in most of the experiments moreover it takes less time.

ÖZET

REKABETÇİ VE HİYERARŞİK TESİS YERİ SEÇİMİ PROBLEMİ İÇİN YENİ SEZGİSEL YÖNTEMLER

Tesis yeri seçimi probleminin türevleri üzerine uzun yıllardır çalışılmıştır. Bu çalışmada iki türevin karması ele alınmaktadır: Rekabetçi tesis yeri seçimi ve hiyerarşik tesis yeri seçimi.

Bir firma halihazırda bir rakibinin hiyerarşik tesislerini yerleştirmiş olduğu bir pazar ortamına, hiyerarşik tesislerini yerleştirmek istiyor. Amaç toplam kapılan pazar payından zincir oluşturmak için gerekli toplam maliyet çıkarılarak elde edilen toplam net karı en büyüklemektir.

Hiyerarşik yapı, sonrakini içeren servis hiyerarşi modelidir ve tüm tesislerin iki seviyeli olduğu varsayılmaktadır.

Ana varsayımımız bir müşterinin istemini (satın alma gücünü) iki zincir arasında, her zincirin ona en yakın tesisi için çekicilik düzeyi ile orantılı olarak bölüştürdüğüdür. Bu müşteri olma davranışı, olasılıksal ve gerekircii müşteri olma davranış modellerinin bir karmasıdır.

Bu çalışmanın katkısı tesislerin çekiciliğinin sürekli bir karar değişkeni olarak ele almasıdır. Aynı zamanda her düzeyde açılacak olan tesislerin sayıları önceden belirlenmiş değildir.

Bu problem için bir tamsayılı doğrusal olmayan model geliştirilmiştir. İlk olarak GAMS v22.0'ın içinde yeralan SBB çözücüsü uygulanmıştır. Daha sonra bir tavlama

benzetimli algoritma geliştirilmiş ve bu algoritmanın içinde herbiri farklı ekle kaldır ölçütlerine sahip olan iki farklı strateji uygulanmıştır. İlk strateji (SASCAD) komşuluk aramak için ikinci en yakın ekle kaldır ölçütlerini barındırırken, ikinci strateji (SARAD) rasgele ekle kaldır ölçütlerini kullanmıştır. Daha sonra bu iki strateji karşılaştırılmış ve SASCAD'ın daha iyi sonuç verdiği fakat daha yavaş olduğu görülmüştür. SARAD çoğu deneyde SASCAD ile aynı sonuçları daha kısa sürede yakalamayı başarmıştır.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

A_{ik}	The attractiveness of a level-k facility that is located at potential site i
A_k	The common attractiveness of all level-k facilities
b_{cjk}	Allocation parameter of competitors facilities
BP_{jk}	The buying power of customer j for type-k demand
CA_{ck}	The attractiveness of level-k facility c that belongs to competitor
CA_k	The common attractiveness of all level-k facilities that belong to competitor
cd_{cj}	Euclidean distance between customer j and competitor's facility c
$coef$	A predetermined coefficient used to update the temperature
d_{ij}	Euclidean distance between customer j and facility located at site i
f_k	Fixed cost of opening a level-k facility
F_v	Fibonacci number
i	Index of sites
$impro\_crt$	Improvement criterion
$intem$	Initial temperature
j	Index of customers
k	Index of levels and types
m	Total number of customers
$\max\_initial$	Number of times the algorithm is run
$\max\_iter$	Total number of iterations made in an improvement loop
$\max\_no\_imp$	A predetermined parameter used for termination
n	Total number of sites
$NL1$	Number of competitor's level-1 facilities
$NL2$	Number of competitor's level-2 facilities
o	Index of occupied sites
$o_{\min}$	Index of the occupied site to be dropped

OS	The set of sites where there are facilities already located
P_{ijk}	The probability (frequency) that a customer j will patronize one's own chain's facility located at i for demand type-k after he\she is allocated to facility located at site i
P_{jk}	The probability or frequency that a customer j shops for type-k demand from one's own chain
Pf_{tjk}	The probability (frequency) that customer j shops for type-k demand from potential site t
q	Iteration counter in Fibonacci search
r_u	The objective value of the solution when a facility is temporarily added to an unoccupied site u
s	Current objective function
u	Index of unoccupied sites
$u_{\max}$	Index of the unoccupied site to be added
ua_k	Unit attractiveness cost of a level-k facility
US	The set of empty sites
v	Total number of observations
w	A predetermined coefficient used to determine the initial interval in simplex search
x_{ik}	Location variable
y_{ijk}	Allocation variable
$z_{\emptyset}$	The objective value of the solution when a facility at the potential site o is temporarily deleted
α	Reflection parameter
β	Contraction parameter
γ	Expansion parameter
Δ_a	Difference between the objective values of new and existing solutions
Δ_{avg}	Average difference between the objective values of new and existing solutions
ϵ	Distinguishability constant of Fibonacci search
ζ	The final length of the interval in Fibonacci search

λ	Parameter for sensitivity of distance
ϖ	Total number of observations in Fibonacci search
ρ	Tolerance number in Simplex search
σ_q	Lower bound of the interval of Fibonacci search for q^{th} iteration
υ	Number of unoccupied sites
Ψ_a	Objective value for an existing solution
Ψ'_a	Objective value for a new solution
ψ_q	Upper bound of the interval of Fibonacci search for the q^{th} iteration
Ω	Coefficient used for the initial interval in Fibonacci search

BSSS	Big square small square
COHILP	Competitive and Hierarchical Location Problem
H	High
L	Low
MAXCAP	Maximum Capture problem
MCI	Multiplicative Competitive Interaction model
MSM	Market Share Model
SA	Simulated Annealing
SARAD	Simulated Annealing with Random Add Drop
SASCAD	Simulated Annealing with Second Closest Add Drop
SBB	Simple Branch and Bound
SCMS	Single facility Competitive Maximum Capture
SMC	Single facility Maximum Capture
TCA	Total attractiveness cost of constructing a chain
TCMS	Total captured market share
TFC	Total fixed cost of constructing a chain
TNP	Total net profit

1. Introduction

Facility location problems deal with finding the optimum locations of a number of facilities with various objective functions such as minimizing the demand-weighted sum of distances between facilities and customers. These problems have been studied for a long time. Presently, researchers from many different fields like management science, computer science, mathematics, architecture, and economy focus on various interpretations of facility location problems. When the types of facility location problems are investigated, it is possible to meet a large number of different versions. This thesis considers two versions: the Competitive Facility Location Problem and the Hierarchical Facility Location Problem.

Competitive facility location aims to maximize the captured market share of a certain chain where there are other competitor(s) that already entered the market or soon will enter the market. Within the competitive facility location models there are also many different approaches with respect to the kind of competition, the kind of demand and patronizing behavior. In most of the studies, the locations of facilities are assumed to be unknown, while there are only a few works that investigate also the attractiveness of the facilities.

In the literature, the competitive facility location is considered mostly in private sector models such as supermarkets, fashion store chains,and franchise systems, with an objective of maximizing profit.

There is a hierarchical structure within the facilities in terms of services offered to customers and one wants to locate these hierarchical facilities considering different objective functions such as maximum coverage, and minimum travel cost in the hierarchical facility location problems.

The hierarchical facility location is taken into account in public sector models such as health-care systems, education systems, emergency medical service systems, and telecommunication systems.

These two different models can be combined considering hierarchical facilities which are planned to be located in a competitive environment. However, in the literature there is only one paper analyzing this combination. Considering hierarchical supermarkets, hierarchical technology markets, banks, and private hospitals there is a huge need for competitive and hierarchical models, applications, and algorithms.

In this thesis we deal with a competitive and hierarchical facility location problem in which there are competitor's facilities already existing in the market area and one wants to locate a number (not predetermined) of hierarchical facilities to form a chain so as to maximize the net profit. Net profit is calculated by subtracting the fixed costs of facilities and costs of attractiveness from the total captured market share. The location space is discrete which means there are potential sites some of which will be chosen as facility locations. Competition is static, hence competitor will not react to any changes in the environment. Demand is inelastic so it does not change with respect to the distance between competitor and facility, number of facilities existing or the utility function of customers. The patronizing behavior is a hybrid of deterministic and probabilistic approaches. A widely used Huff-like gravity model is employed once a customer is allocated to one's own chain's closest facility and competitor's closest facility. In other words, a customer splits his/her demand among two facilities (each belongs to a different competitor) proportional to the attraction level to the closest facility of each chain. It is the first time such a patronizing behavior is being utilized. From the hierarchical point of view, the model is a "successively inclusive facility location model" where there are two kinds of facilities. First level facility only offers type-1 service, whereas a second level facility can provide both type-1 and type-2 services (Daskin, 1995). It is the most appropriate hierarchical structure to represent the real life situation of competitive and hierarchical facilities. More than having a rarely studied structure, this thesis differs from the other studies existing in the literature with respect to decision variables. Both the facility locations and their attractiveness values are treated as decision variables. In the literature, few papers treat attractiveness as decision variables. Furthermore, most of these studies use a discrete set of attractiveness values or solve a single-facility location problem, whereas our model attempts to find continuous attractiveness. Considering attractiveness as a continuous decision variable makes the model more difficult since optimization with respect to location and attractiveness should be carried out simultaneously. Moreover the customer's proportion of demand is affected by both the location and attractiveness that is why the model catches a trade-off between location and attractiveness.

Another feature of this study is that: the number of facilities to be located is not predetermined, but is also a decision variable. It is decided by the solution of the model with respect to trade-off between fixed costs of the facilities and the extra profit from these facilities. Similarly there is no predetermined interval to restrict the attractiveness values which are determined by the "attractiveness costs".

We develop a mixed integer nonlinear programming model for this problem. It is known that global optimal solution is not guaranteed by using any exact solution technique. After randomly generating some instances, we first employ the Simple Branch and Bound (SBB) solver within GAMS suite v22.0 to solve the generated instances. Then Simulated Annealing (SA) algorithm, with add-drop and swap moves is developed. Results show that using the proposed algorithm is very efficient to find near optimal solutions.

2. Literature Review and Background

2.1. Competitive Facility Location

The competitive facility location problem aims to optimally locate facilities while there are other facilities in the market that belong to a competitor. These problems have been studied for a long time and studies mainly focus on the private sector with the aim of maximizing the total captured market share of a chain. Hotelling's (1929) paper is agreed to be the first paper about competitive facility location. He considers the location of two competing ice-cream vendors on a line and assumes that the customers patronize the closest facility which is called "proximity assumption" and the buying power is uniformly distributed on the line. Because of the proximity rule he use, all facilities are assumed to be equally attractive and when such a situation exists it is sensible to use a Voronoi diagram to divide the region as Hotelling did.

He achieves simple but efficient results such that when a company A locates a facility to the right side of the line, the competitor company B will definitely locates its facility to the left side of the line, but has a tendency to locate its facility close to A. This act will assist company B to capture more customer than A. However some customers that are close to left edge of the line will have to travel a long distance. Moreover if another competitor company C decides to locate a facility then the closeness of company A and B will be an advantage for company C. He also suggests that different attributes than distance can be used as attractiveness attributes. However he did not focus on them to simplify his model.

Competitive facility location models can be classified with respect to various features it accommodates. In the following section we will give an extensive classification.

2.1.1. Locational Space

2.1.1.1. Continuous Space Models. These models assume that the optimal locations for facilities can be anywhere in the plane. Starting with Hotelling (1929), many papers that employ continuous solution space have been written so far. In most of these studies it is argued that having a finite set of possible locations may prevent to find the optimal solution since the optimal solution may not be included in that set.

Drezner and Drezner (2006a, 2006b), Drezner (1994, 1998), Drezner et al. (2002) use continuous space, and they usually develop a gravitational Huff-based formulation. McGarvey and Cavalier (2005) also study continuous location space but propose that there are restricted regions, on which location of a facility is forbidden with respect to travel, location or both criteria.

Usually Weizsfeld-like algorithms (Weizsfeld, 1936) and Big Square Small Square (BSSS) techniques (Hansen et al., 1985) are employed to find optimal or near-optimal solutions in continuous space when the problem is a single-facility location problem. For multi-facility versions, Weizsfeld or BSSS algorithms are embedded in a univariate search procedure. We have to emphasize that Weizsfeld algorithm is a local search algorithm and one who wants to find global optimal solution may begin from multiple initial solutions. Fernandez et al. (2007) propose a new partitioning method with pruning by an adaptive multi-section rule. Drezner and Hamacher (2002) is a good reference in terms of utilizing different heuristics in continuous space and observing differences between them.

A more comprehensive survey about competitive continuous facility location problems can be found in Drezner (1995).

2.1.1.2. Network Models. In network models the facilities are allowed to be located anywhere along the edges and vertices of the network. Wendell and McKelvey (1981) is known to be the first who carried the competitive facility location problem to a network as an extension of Hotelling (1929). They intend to answer the question: "When can and when cannot a firm choose a facility location that will guarantee it at least as many customers as its competitor, regardless of where its competitor locates?" They call the location which satisfies that question, "a locational equilibrium". Their final observation is that very severe symmetry requirements on individual locations and also on demand functions are necessary to capture the equilibria.

Hakimi (1983) also analyzes the competitive facility location problem on the network and proves that one can find a set of optimal locations on the vertices of the network under certain circumstances.

2.1.1.3. Discrete Space Models. It can be seen that the discrete space model is a special case of network models. If we retain the edges of the network to be potential locations of the facilities, then the model simply becomes a discrete space model. In discrete space models there are only a finite set of potential sites and one seeks to choose some of these sites as locations for facilities to be opened to maximize the captured market share.

It may not be a realistic approach to assume that a facility can be located anywhere in the plane since in real life situations it is not easy to find huge lands required to locate especially large-scale facilities. That is why using a discrete space model makes sense in a real-life competitive environment.

The first paper about discrete competitive location is by Goodchild's (1984) where a location-allocation market share model (MSM) is presented. It is an extension of Hotelling's approach. In MSM, the solution space is not a line but there are some user provided potential locations from which optimal or near optimal locations will be chosen as facility locations by using heuristic algorithms.

A p-median problem with gravitational model is adopted in discrete space by Drezner and Drezner (2007) who apply the steepest descent method and a Tabu Search

(TS) heuristic. Berman and Krass (2002) also use a greedy algorithm to solve a discrete space problem with elastic demand and claim that greedy type algorithms are efficient to obtain high-quality approximate solutions. It is observed that discrete space models are more popular in duopolistic markets particularly if the price decision is also taken into account. Such a problem is observed by Fischer (2002). Another paper on discrete space competitive facility location problem is MAXCAP which develops different models based on various competition assumptions. (Revelle, 1986) Extensions of the MAXCAP problem and more information about discrete space models can be found in Serra and Revelle (1995).

2.1.2. Patronizing Behavior

Customers behave differently when they choose a facility to visit. This difference may occur due to the population in the area, distribution of income and sector. For instance, for a customer looking for a luxury good, price, quality of the product, transportation, product variety and distance are some of the important attributes each having a different weight. For a customer shopping essential goods such as bread, sugar etc. distance has a more dominant weight and some of the other attributes mentioned above may not be important. A customer may prefer to patronize only one facility or distributes his/her income among some facilities. All of these preferences constitute a patronizing behavior.

When there is a competition, decision makers firstly have to predict the market share they can capture in order to locate the facilities optimally. This, in turn, strongly depends on the patronizing behavior of the customers. If one employs different patronizing behavior assumptions to a competitive model, he/she will definitely observe that the captured market shares will significantly differ from each other. For this reason, while forming a mathematical model for competitive facility location problem, the decision maker has to decide on the behavior of customers. We can basically divide patronizing behavior assumption into two classes: Deterministic patronizing behavior and probabilistic patronizing behavior. For a comprehensive survey one can see Drezner and Hamacher (2002).

2.1.2.1. Deterministic Patronizing Behavior Models. Deterministic models assume that customers patronize a facility with probability one with respect to some criteria they take into account. These criteria may change for different aspects, sectors or the elements that the model includes.

The first paper using deterministic patronizing behavior belongs to Hotelling (1929) where he assumes that customers patronize the closest facility according to Euclidean distance between a customer and a facility. This assumption is called "proximity assumption" and ensures that customers located at the same demand point will patronize the same facility, which is named "all or nothing property". Here the criterion customers focus on is only the closeness of the facility. In MAXCAP, Revelle (1986) also adopts proximity assumption by adding an equal distance rule: if two facilities are equally close to a customer, then this customer splits his/her demand into two and apportions between two competing facilities. Similar to these two papers, Plastria and Carrizosa (2004) also assume a single patronizing criterion which is the prices offered by facilities. Customers patronize the facility which offers them the lowest price while the travel cost of a customer is paid by the facility he/she patronizes.

Drezner and Drezner (1994) extend this assumption by incorporating attributes other than distance to their model, which is referred to "deterministic utility model". Each customer has a utility function and while choosing a facility to patronize, customers behave in a manner such that their utility function (satisfaction) is maximized. In some models this utility function is additive as in deterministic utility model. In others it is multiplicative as in Multiplicative Competitive Interaction Model (Nakanishi and Cooper, 1974). In the deterministic utility model, the utility function is a weighted sum of attributes x_p, $p = 1, 2, ...m$. Utility function can be represented as:

$$U = \sum_{p=1}^{m} w_p f_p(x_p) \tag{2.1}$$

where

$$
\begin{aligned}
x_p &= \text{attributes,} \\
w_p &= \text{associated weight of } x_p, \\
f_p(x_p) &= \text{a function of } x_p.
\end{aligned}
$$

Here, all the attributes of existing and new facilities are assumed to be known except for the distances of the new facilities. To characterize this idea we can give some examples for attributes. One attribute can be facility size, the other may be attractiveness of facilities or prices that facilities offer. As mentioned earlier these attributes may change due to necessities of the model but the distance factor will definitely be included as an attribute. They also propose a new concept called "break-even distance". It is the maximum distance a customer is willing to travel to a new facility. Assume that there is an existing facility i in the area and we want to locate a new facility. Since all the attributes of the existing facility are predetermined, we can easily calculate its utility function for a customer and let's call it u_i. We also know all the attributes of the new facility except for the distance so the utility function of the new facility is a function of the distance which is represented as $u(d)$. A customer will patronize the new facility rather than the existing one if $u(d) > u_i$. Therefore the maximum distance he/she would like to travel for the new facility is the solution of $u_i = u(d)$

Since $u(d)$ is a decreasing function of d, this equation has a unique solution. Likewise, if there are a certain number of facilities that have already been located, the break-even distance for a demand point is the minimum of the break-even distances that have been calculated for all existing facilities.

There are also other utility functions used in a deterministic model. In Aboolian et al. (2007a) customers patronize the facility with the maximum utility function but this time utility is a function of price, attractiveness and distance.

However, none of the above models could overhold the "all or nothing property" although it is not a realistic approach since in reality not all the customers from a demand point patronize the same facility. As a result, we have the probabilistic patronizing behavior models.

2.1.2.2. Probabilistic Patronizing Behavior. The first paper which introduces a probabilistic patronizing behavior for the competitive facility location problem mainly based on Reilly's (1929) "law of retail gravitation" is the gravitational model by Huff (1964). He put forward very important results from empirical studies: The proportion of customers attracted by a facility is strongly affected by the distance and the facility area which means that the probability (frequency) that a customer patronizes a facility is proportional to the facility area and inversely proportional to a power of the distance function. By taking these factors into consideration, Huff proposes a formal expression of the model as:

$$P_{ij} = \frac{\frac{S_i}{T_{ij}^{\lambda}}}{\sum_{q=1}^{n} \frac{S_q}{T_{qj}^{\lambda}}} \tag{2.2}$$

where

P_{ij}	=	the probability that a customer at demand point j travels to a particular facility i,
S_i	=	the size of facility i,
T_{ij}	=	the travel time from a customer's travel base j to a facility i,
λ	=	a parameter which is to be estimated empirically to reflect the effect of travel time on various kinds of shopping trips.

It is stated that if λ goes to infinity, the customer patronizes the closest facility with probability 1. Therefore Huff's model is a generalization of the proximity assumption (Drezner 1995). Huff defines the expected number of customers at a demand point j that are likely to patronize facility i as:

$$E_{ij} = P_{ij} C_j \tag{2.3}$$

where

E_{ij} = the expected number of customers at demand point j that are likely to patronize facility i,

C_j = the number of customers at demand point j.

Finally, he obtains

$$\Gamma_i = \sum_{j=1}^{n} P_{ij} C_j \tag{2.4}$$

where

Γ_i = the trading area of a particular facility i, that is the total expected number of customers within a given region who are likely to patronize facility i.

Here we have to emphasize that all parameters except for T_{ij} (the travel time involved getting from a customer's travel base j to a facility i) are predetermined and the only decision variable is the location of the new facility.

In (2.4) if we replace C_j with the total buying power of the customers who are located at demand point j, then we obtain "captured market share of facility i". It is clear that captured market share, a widely used term comes from Huff's formulation.

Huff's model is simple and although he considers a single facility, the model is very efficient and served as foundations for many competitive location models. A large number of studies on competitive location benefited from his model. All criticisms on Huff's model compromised that using a single attractiveness measure (facility floor area) is restrictive and unrealistic hence an extension is needed. In this way, Nakanishi and Cooper (1974) introduce a new model called "Multiplicative Competitive Interaction Model" (MCI) which also considers a single facility. The idea is similar but the utility model that they employ is multiplicative which contains multi-attributes.

$$\pi_{ij} = \frac{\prod_{k=1}^{q} x_{kij}^{\beta_k}}{\sum_{t=1}^{m} \prod_{k=1}^{q} x_{kit}^{\beta_k}} \tag{2.5}$$

where

π_{ij} = Probability that a customer in the j^{th} demand point patronize facility i,

x_{kij} = The k^{th} variable (attribute) describing facility i at demand point j,

β_k = parameter for sensitivity of π_{ij} with respect to attribute k.

In this model, all attributes except for the location of the new facility are predetermined and the only decision variable is the location of the new facility. If the number of attributes (used for representing the attractiveness) is greater than one, generally additive or multiplicative utility functions are adopted. Drezner and Drezner (1994, 1996) make use of additive utility functions in random and in deterministic utility models. The main difference between multiplicative and additive functions is that in additive models absence or scarcity of an attribute can be compensated by increasing the value of another attribute or attributes, which means that it is not important not to include one or more attributes in the model. However, in multiplicative models if an attribute of a facility is set to zero, then the probability that a customer patronizes that facility will definitely be zero. It simply points out that each attribute is indispensable for the customer and in the absence of an attribute the customer will not patronize that facility.

The MCI model is more complex in terms of facility attributes but its main problem is the estimation of the parameters. Moreover, using binary attributes like 0-1 or present-absent is impossible since estimating parameters requires to take the logarithms of the ratios between the attribute values and their geometric means, and the use of binary values makes the geometric mean zero. To overcome this problem Mahajan et al. (1978) make transformations over the MCI model.

Huff (1964) and Nakanishi and Cooper (1974) study on a single facility problem. Achabal et al. (1982) extend the MCI model to the location of multiple facilities which belong to one's own chain. They solve this nonlinear integer programming problem by using a random search procedure with an interchange heuristic to obtain optimal or near-optimal solutions. As we have already pointed out, a large number of studies that focus on competitive facility location utilize Huff's gravitational model. Some of them make small modifications on Huff's model to be able to adapt it to their specific problems.

Drezner (1994) solves a gravity-based single facility location model in continuous space by applying Weizsfeld algorithm. It is claimed that the objective function of the problem is not concave and includes local optimal solutions. Since Weizsfeld algorithm is a local search procedure it is advised to use different initial solutions in order not to get stuck with the local optima. Attractiveness of the facility is predetermined and by performing sensitivity analysis for different values of attractiveness, Drezner reaches different results in optimal locations with respect to changes in attractiveness values. A generalization of this study is proposed in Drezner (1998) by adding a limited budget constraint and letting multiple facility location. She develops two models one of which has a fixed budget while the other includes flexible budget. Drezner performs a two-phase procedure to overcome the "simultaneous location of several new facilities anywhere in a continuous plane problem". First phase is to extend the single facility case to the location of multiple facilities with known values of attractiveness. The second phase is the formulation of allocating the resource among the new facilities where there is a predetermined budget and decision variables represent the portion of budget planned to be spent to each new facility. She attempts to find optimal locations, optimal resource allocation policy, and optimal number of chain facilities to be located. Fixed budget model is solved by a univariate search algorithm that includes Weizsfeld whereas flexible budget model is handled by adding a steepest descent phase to the univariate search algorithm. This paper is the first one which investigates the optimal allocation of budget among new facilities. However the budget constraint enforces to spend all the budget given which may bear worse results than not spending the whole budget.

After this study Drezner et al. (2002) practise the same Weizsfeld algorithm that used in Drezner (1994). A criteria-based algorithm, an ascent algorithm, a SA algorithm and a SA and local search algorithms are employed to solve the multi-facility problem. They compare the results of these algorithms and conclude that the best algorithm for the problem is the SA and local search algorithms.

Another paper that demonstrates a gravity-based multi-facility model in continuous space also brings out a budget constraint and also introduces a capacity constraint and a forbidden region constraint (McGarvey et al., 2005). Here the criticism we made for Drezner (1998) is not valid because in this model the budget constraint is "less than or equal to" type and one does not have to spend the whole budget that is dedicated. They employ BSSS algorithm embedded in a cyclic search heuristic and they use Drezner et al.'s (2002) SA algorithm to obtain initial solution.

The gravity p-median model in a discrete space is heuristically solved by a steepest descent algorithm and a tabu search algorithm in Drezner and Drezner (2007). Then, these two algorithms are compared and it is seen that tabu search algorithm is more efficient than the steepest descent algorithm. Fernandez et al. (2007) and Aboolian et al. (2007b) are other papers which modified Huff's gravity-based formulation for their models.

A different probabilistic approach to remove "the all or nothing property" is proposed by Drezner and Drezner (1996) by composing a random utility model which is very similar to the deterministic model. In that model the attributes, their weights, and distances are drawn from a probability distribution with known means and variances. They try to solve the problem using a Weizsfeld-like algorithm. However, the objective function by its nature includes a k-dimensional integral so the computational effort is great to find the best location. Therefore in Drezner et al. (1998) they define a simplified random utility model which is approximated by a logit function.

Conversion of random utility functions into gravity-based model is a third approach to demolish the "all or nothing property" (Benati and Hansen, 2002). The random utility function of a customer i for a facility j is given as :

$$\tilde{u}_{ij} = a_i - \beta d_{ij} + \tilde{\epsilon}_{ij} \tag{2.6}$$

where

$\tilde{u}_{ij}$	=	the total utility of the choice of the facility i by the customer j,
a_i	=	the average attractiveness of facility i,
$\beta \geqslant 0$	=	a number which represents how people discount distances,
d_{ij}	=	the distance between i and j,
$\tilde{\epsilon}_{ij}$	=	the random part and it stands for nonobservable variables (described by some joint function from which all customers are assumed to draw).

They claim that the probability of customer j patronizes facility i is equal to the probability that the utility function of facility i for the customer j is the maximum utility function among the utility functions of all facilities for customer j. It is interesting that if one assumes that the random part of the utility function is independently and identically distributed with the Weibull distribution, then the probability becomes

$$P_{ij} = \frac{\exp(a_i - \beta d_{ij})}{\sum_{k \in V'} \exp(a_k - \beta d_{kj})} \tag{2.7}$$

where V' represents the set of located facilities.

We see that the resulting model is a gravity-based model.

2.1.3. Customer Demand

In the literature competitive facility location problems generally assume that the demands (buying power of the customers) are predetermined and constant, which is known as "inelastic demand". Especially in essential goods which are generally sold in supermarkets or grocery stores, it is reasonable to assume inelastic demand since it is not meaningful for customers to increase or decrease their demand with respect to distance, quality or price for these vital goods.

However, for luxury goods demand may be highly sensitive to supply or price, and one can form a demand function so that demand reacts with respect to changes in price, distance etc. (Plastria, 2001). This kind of demand is called "elastic demand". For absorbing the reason of utilizing elastic demand, we would like to define some keywords which are mostly used in marketing literature.

Cannibalization occurs when new facilities capture some of the demand from existing facilities. Capturing the customer demand of competitor's facility is beneficial, whereas cannibalizing the demand from facilities belonging to one's own chain will not improve the objective value. Conversely, it may cause undermining the profit of pre-existing facilities (Berman and Krass, 2002).

Market expansion is the increase in customer demand when the facilities increase their score of service. This score may increase as a result of new facilities being added, through design improvements or price adjustments. The relationship between market expansion and cannibalization highly affect the profitability. For example, if market expansion outweighs the cannibalization effect, a pre-existing facility might capture more demand after new facilities are added (Aboolian et al., 2007b).

Fischer (2002) proposes a two-stage model for duopolistic competition and employs a demand function of price. ($D_i(p_i)$ is the demand of market i and p_i is the price which market i offers). He determines the upper bound and lower bound of the price by setting $D_i(p_{i,\max}) = 0$ where $p_{i,\max}$ is the upper bound and a facility that does not want to serve market i may set its price to $p_{i,\max}$. $D_i(0) = D_{i,\max}$ where 0 is the lower bound and $D_{i,\max}$ is the maximum demand of market i.

A demand function of utility which is also a function of facility capacity and distance is assumed in McGarvey and Cavalier (2005). Another utility based demand function is investigated in Aboolian et al. (2007a) where demand is a function of price, facility attractiveness, and travel cost. In Aboolian et al. (2007b), they use a similar function which does not include price. A more realistic approach is handled by Berman and Krass (2002) where they avoid using the term "elastic demand" and call it "non-constant expenditure function". It is stated that the expenditure function is assumed to be affected by both the location and the number of the facilities. With this framework they capture both the market expansion and cannibalization terms.

Drezner and Drezner (2006b) bring a different perspective to the demand elasticity and propose that there is a constant buying power which a customer plans to spend for a branch of a sector. However, if the facilities working in that branch are too far, then the customer may choose not to spend his/her whole buying power. The total buying power will be spent if the facilities are very close to customers. But if they are not close enough, then a portion of the buying power will be spent. The model has two objective functions. The first one is the maximization of the buying power spent at all competing facilities and the second is the maximization of the buying power captured by one's own chain.

The probability P that a customer at site j will not spend his/her buying power at facility i is

$$P = 1 - e^{-\lambda_i d_{ij}} \tag{2.8}$$

where

λ_i = parameter for sensitivity of distance,

d_{ij} = distance between customer j and facility i.

The probability that a customer will not spend his/her buying power at any facility is $\prod_{i=1}^{k}[1 - e^{-\lambda_i d_{ij}}]$. And the total buying power spent at all facilities is:

$$B_j(1 - \prod_{i=1}^{k}[1 - e^{-\lambda_i d_{ij}}]) \tag{2.9}$$

Finally the total buying power at all communities spent at facility i is formulated as:

$$M_i = \sum_{j=1}^{n} B_j (1 - \prod_{r=1}^{k} [1 - e^{-\lambda_r d_{rj}}]) \frac{e^{-\lambda_i d_{ij}}}{\sum_{r=1}^{k} e^{-\lambda_r d_{rj}}} \tag{2.10}$$

It is seen that the first part of equation (2.10) is the available buying power and the second part is a Huff-like gravity based model.

Drezner and Drezner (2006b) first form a single facility maximum capture (SMC) model without competition and in solution procedure it is aimed to maximize the captured market share which could not be captured by any other facility before. Then they modify the model to comprise the competition (SCMC). Both of these are continuous location models, and they are solved by Big Triangle Small Triangle algorithm which is a version of BSSS. As a consequence of the lost demand, the total buying power captured according to SCMC model is lower than the total buying power captured according to the Huff model.

2.1.4. Attractiveness Assumption

Attractiveness is a concept which represents the competitive advantage of a facility. The word has a broad meaning. Some papers use this word instead of an attribute which describes a feature of the facility that affects the customer decision. That attribute can be the facility area as in Huff's model (1964), the quality of the product or the design of the facility (Fernandez et al., 2007), the price, the capacity of the facility (McGarvey and Cavalier, 2005). However, in some other studies it is possible to see that attractiveness is defined as a combination of some attributes (Drezner and Drezner, 1994, 1996), (Nakanishi and Cooper, 1974). If attractiveness is a combination, then one can investigate all the attributes of this combination or may just declare that attractiveness measure is assumed to be a combination but does not deal with the attributes one by one.

One of the important issues is whether the attractiveness is a decision variable or a parameter. In the literature many papers assume attractiveness as predetermined, whereas there are only a few papers that treat attractiveness as a decision variable. Utilizing attractiveness as a predetermined parameter will simplify the model. But the attractiveness values should be chosen carefully in order to solve the problem realistically. The objective value will significantly be affected with respect to attractiveness. If the attractiveness values are not realistic, then the solution will not be useful for a real life situation. Drezner (1994) applies a sensitivity analysis for different attractiveness values in a gravity based single-facility location problem, and concludes that the market share captured by the facility is sensitive to both facility location and attractiveness.

As mentioned before, little research has been done on the determination of the attractiveness. The first paper which treats the attractiveness as a decision variable and involves a simultaneous optimization of location and attractiveness in continuous space is by Plastria (1997) and a broad generalization of it is made in Plastria and Carrizosa (2004) which is a single facility model in a continuous space. Customers patronize the facility to which they are attracted the most, which is known as a maximum utility deterministic model. They state that the solution of the problem can be reduced to a bicriteria maxcovering-minquantile problem for which solution methods are known. Drezner and Drezner (1994) also develop a deterministic model with the aim of optimally determining both location and attractiveness. Another single facility location problem with gravity based model is intended to be solved by using a Weizsfeld-like algorithm (Fernandez 2007). They also employ an interval branch and bound algorithm which progress by the subdivision of boxes.

The paper by Eiselt and Laporte (1989) and Santos et al. (1998) answer the question of how to determine both the location and attractiveness of new facilities in a discrete space with gravity based model. Their main contribution is to consider the attractiveness as a continuous decision variable. Eiselt and Laporte (1989) use the term "weight" for attractiveness. They investigate the simultaneous optimization of location and weight of a single facility on a network where user provided nodes are defined as candidate locations for the new facility. A huff-like patronizing behavior is adopted. The problem is a special case of r/xp medianoid problem where $r = 1$ and weight is a continuous decision variable. Objective function consists of captured market share and cost of opening a facility with an unknown weight. It is given as

$$\max\ z = \sum_{i \in M_2} (\sum_{j \in N} b_j Y_{ij} - cw_i) \tag{2.11}$$

where

j = index of customers,
i = index of nodes,
M_2 = set of unoccupied nodes of network,
N = set of all nodes of network,
b_j = demand of customer that is located at node j,
c = unit cost of weight,
w_i = weight of the new facility that is located at i.

and

$$Y_{ij} = \frac{w_i x_i / d_{ij}}{w_i x_i / d_{ij} + \sum_{k \in M_1} \bar{w}_k / d_{kj}} \qquad j \in N,\ i \in M_2 \tag{2.12}$$

where

d_{ij} = shortest distance between node i and node j,
M_1 = set of occupied nodes of network,
$\bar{w}_k$ = known weights of already existing facilities.

$$x_i = \begin{cases} 1 & \text{if the new facility is located at node } i \\ 0 & \text{if not} \end{cases}$$

Obviously it is seen that most of these papers investigate a single facility or p-median facility location problem for simplification since attractiveness as a decision variable causes a great complexity in the solution procedure.

Aboolian et al. (2007b) challenge the determination of both location and attractiveness and also relax the constraint of fixed number of facilities. The model is a discrete space, elastic demand, gravity-based type and seeks the answer for the following questions:

1. How many facilities should be located?

2. Where should these facilities be located?

3. What kind of facilities (in terms of size, product variety, and other design aspects) should be located?

They admit that the number of facilities to be located is affected by the attractiveness of facilities since fewer more attractive facilities may be used instead of less attractive facilities. They provide some discrete attractiveness scenarios to employ in the solution procedure hence restrict the attractiveness values. A weighted greedy search heuristic is proposed which starts with an empty set, then identifies the location-attractiveness pair that yield the largest improvement and stops when no improving location-attractiveness pair are available.

2.1.5. Competition

When there is a competition in the area it is sensible to expect some future reactions from the competitors after making a move. Since competitive facility location generally deals with the location of facilities, expecting a change in the location of competitor's facilities is almost impossible because it will be too costly to open a new facility or to relocate a facility. Therefore in most of the papers mentioned above the competition is assumed to be static which states that a competitor cannot respond to moves of one's own chain. Hence the characteristics of the competition are known and assumed to be fixed.

In a dynamic competition environment, a competitor can relocate its facilities, close its facilities, open some new facilities, modify the attractiveness values of its facilities etc. Revelle and Serra (1991) propose a location-relocation model (MAXRELOC) where both location of new facilities and relocation of existing facilities are allowed. Even when the dynamic competition is assumed in competitive facility location problem, the respond of the competitor generally relies on price adjustments which will be more beneficial for economists (Fischer, 2002).

2.2. Hierarchical Facility Location

The hierarchical facility location problem is a special case of facility location problems where there is a hierarchy between facilities. Assume that level-k facility is the highest level facility and level-1 facility is the lowest level facility while level-0 represents the demand points or customers. Generally hierarchical facility location is faced in public facility planing since most of the public facilities are naturally hierarchical.

These problems differ from each other with respect to flow pattern, spatial configuration, objective function and service varieties.

2.2.1. Flow Pattern

Flow pattern models are usually seen in production-distribution systems. There are two types of flow patterns: single-flow and multi-flow. In single-flow models the flow should start from level-0, go through all levels increasingly and stop at the highest level or vice versa. There is no restriction that the flow should start from level-0 for multi-flow models. The flow can start from any lower level-m to any higher level-n where m,n $\in\{0, 1, 2, ..., k\}$. However multi-flow models are more complex than single-flow models with respect to location decision. For more information about multi-flow models one can see Serra and Revelle (1993).

2.2.2. Spatial Configuration

2.2.2.1. Coherent Systems. Coherent systems restrict the model so that customers that are assigned to the same lower level facility, have to be assigned to the same higher level facility later. Thus coherent systems provide single sourcing to satisfy the demand in facility location problems with capacity constraint.

2.2.2.2. Non-coherent Systems. The single source constraint is relaxed in non-coherent systems. For a customer, a higher level-m facility which will satisfy his/her type-m demand, is chosen independently from the lower level allocations of that customer.

2.2.3. Objective Function

Hierarchical facility location models are widely used in different fields such as health-care systems, solid waste management systems, production and distribution systems, education systems, emergency medical services, telecommunication networks etc. Hence various objective functions are seen in the hierarchical facility location literature due to the field in which the model is developed. Most commonly adapted objective functions are median based models, covering models and fixed charge location models.

The aim of median based model is to minimize the total demand weighted travel costs between customers and hierarchical facilities. In covering models a customer is assumed to be covered if there is a facility which is close enough for that customer. Typically two different objective functions are met in covering models:

(i) Set covering where one tries to minimize the total number of facilities to be opened by ensuring that all the customers in the area are covered.

(ii) Maximum covering which intends to maximize the number of customers who are covered by locating a certain number (predetermined) of facilities. As an example one can see the model in Teixeira and Antunes (2008).

Pirkul and Jayaraman (1996) examine a hierarchical fixed charge location problem with the objective of maximizing total facility fixed cost and travel costs where they also added capacity constraint.

All the studies mentioned above settle the objectives by considering public sector. Generally the aim is to minimize the travel distance or to maximize the number of covered customers whereas our model considers a competitive environment with a target of maximizing the market share captured at each level of buying power of the customers.

2.2.4. Service Varieties

2.2.4.1. A Successively Exclusive Facility Hierarchy. A facility at level-m only serves type-m service and services offered by a level-m facility has no intersection with the set of services offered by a level-n facility where $m, n \in \{1, 2, ..., k\}$ and $m \neq n$. To illustrate the idea an example may be beneficial. The hierarchy of education system simply consists of primary school, middle school and high school. One who would like to get middle school service, can only go to middle school, a high school cannot offer the service she/he demands for.

2.2.4.2. A Successively Inclusive Facility Hierarchy. This hierarchy structure relaxes the constraint above and claims that a level-m facility can offer all services 1 through m to a customer. However this type of hierarchy sometimes may go into division in the application area. A locally inclusive service hierarchy assumes that a level-m facility offers services 1 through m to the customers which have a distance that is close enough to the facility but serves for only type-m demand to the customers which are far away from that facility. A globally inclusive service hierarchy allows a level-m facility to offer services 1 through m without taking the location of the customer into consideration.

Typically bank branches, post offices and some of the health-care systems are appropriate to use a successively inclusive facility hierarchy. It is sensible to employ successively inclusive facility hierarchy for our model which suits the hierarchical gross markets, computer stores etc. In a competitive environment with the objective of maximizing the market share it is unavoidable to employ the successively inclusive facility hierarchy to make a realistic assumption.

Also there are exceptions such as hospitals. In most of these models a facility at level-m serves for some of the demand types between 1 and m, but does not serve for some of the demand types between 1 and m. In such a situation it is sensible to use a hybrid model of the two typical models described above.

For a better understanding of the difference between successively inclusive and successively exclusive facility hierarchy we can examine a basic-median-based hierarchical location formulation.

Inputs:

h_{jm} = demand for type m services at node j,

d_{ij} = distance between node j and candidate location i,

P_m = number of type m facilities to locate.

Decision Variables:

$$X_{im} = \begin{cases} 1 & \text{if a facility of type } m \text{ is located at candidate site } i \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ijm} = \begin{cases} 1 & \begin{array}{l}\text{if demands at node } j \text{ for type } m \text{ services} \\ \text{are satisfied by a facility at candidate site } i\end{array} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \quad \sum_i \sum_j \sum_m h_{im} d_{ij} Y_{ijm} \tag{2.13}$$

subject to

$$\sum_i Y_{ijm} = 1 \qquad \forall\, j, m \tag{2.14}$$

$$\sum_i X_{im} = P_m \qquad \forall\, m \tag{2.15}$$

$$Y_{ijm} \leq \sum_{h=m}^{k} X_{jh} \qquad \forall\, i, j, m \tag{2.16}$$

$$X_{im} = 0, 1 \qquad \forall\, i, m \tag{2.17}$$

$$Y_{ijm} = 0, 1 \qquad \forall\, i, j, m \tag{2.18}$$

In the above model the objective function is the minimization of the demand-weighted total distance. Constraints (2.14) ensure that all the demand types of all customers are satisfied by a facility. Constraints (2.15) stipulate that the number of type m facilities must be equal to P_m. Constraints (2.16) say that type-m demand that originates at node j can be served from a facility at node i if there is a level-m or a higher level facility that is located at i. This constraint represents a globally inclusive service hierarchy model. Note that k is the highest level among the hierarchical structure.

If we replace constraints (2.16) with the constraint:

$$Y_{ijm} \leq X_{ih} \quad \forall\, i, j, m \tag{2.19}$$

then the model will be transformed to a successively exclusive service hierarchy which provides that a type-m demand can only be responded by a level-m facility, not by a facility of type $m+1$ through k (Daskin, 1995).

For a comprehensive overview about hierarchical facility location, see Şahin and Süral (2007). We would like to mention about a few studies on hierarchical facility location problem which have extraordinary models and structures.

Multi-service facility location model intends to co-locate different types of facilities under a single roof in a successively exclusive service hierarchy assumption. The aim is the minimization of the fixed costs of facilities and the travel costs. For instance, in an education system, co-location of pre-school, middle school and high school is examined. The study begins with solving p-median problems for each type of facilities by observing the effect on the travel cost of other type of facilities. Then it is seen that on the optimal solutions some of the locations of one type of facilities overlap with the optimal locations of another type of facilities. This conclusion helps to co-locate multi-facilities on the same location. With a hybrid model which takes all facility types into account, it is concluded that when fixed costs are set to zero, locations of different types of facilities scattered far away from each other whereas when the fixed costs are high enough, all types of facilities tend to be located on the same points. (Suzuki and Hodgson, 2003)

Another study of Suzuki and Hodgson (2004) proposes that facilities are designed in 3 levels where level-1 facility only serves for demand level-1, likewise level-2 facilities offer only level-2 demand, and level-3 facilities satisfy the demand of both level-1 and level-2. Customers are also assumed to vary as ones who desire only type-1 service, ones who desire only type-2 service and ones who desire both type-1 and type-2 service. The last type of customers may be served by a joint facility (level-3) or may prefer to make a trip from a level-1 facility to a level-2 facility or vice versa. This kind of behavior is called multi-purpose trip making. A p-median based model that minimizes the total travel distance and the effect of multi-purpose trip makers is investigated on optimal locations and type of facilities. Results show that services have a tendency to cluster in level-3 facilities, even when the proportion of type-3 customers is small.

The only paper that raises the idea of joining the competitive structure and hierarchical structure in facility location problem belongs to Serra et al. (1992). The objective is to maximize the captured market share. The authors formulate a hierarchical covering model (LOHICO) which allows the location of new facilities and relocation of existing facilities. It is emphasized that most of the hierarchical models in the literature have arisen from classical location formulations such as p-median, the maximal covering location models and no consideration had been given to facilities in a market economy where they compete for customers.

The model includes a deterministic customer behavior in a discrete space where there are specific potential locations for each level of facilities. For instance a level-1 facility can only be settled to a potential level-1 location and the same rule is valid for level-2 facilities. They define a distance-based term T_m which is the additional distance that people are willing to travel to any higher-level facilities for level-m services. Let $S_{jm} = \min(d_m, d_h - T_m)$ where d_m is the distance between customer j and the closest level-m facility of competitor to the customer j, and d_h is the distance between customer j and the closest level-h facility of competitor to the customer j where $h > m$.

If one's own chain opens a level-m facility at a distance lower than S_{jm}, then customer j will fully be captured by that facility. Or in the event of opening a new facility which has a level higher than m, at a distance lower than $S_{jm} + T_m$, again customer i will fully be captured. In the presence of an equation between competitor and one's own chain, the demand of the customer is split into two and it is shared by both competitors. Obviously, it is seen that the only criterion that customers take into account is the distance, no other attractiveness attributes are incorporated. Moreover they fix the number of facilities to be opened and also the number of facilities to be relocated. A programming software is employed to solve the linear mixed-integer programming problem. It is concluded that the model returns successful results even when the competitors optimizes their locations by using a p-median model.

3. Problem Description and Model Formulation

In this thesis we focus on locating facilities and determining facility attractiveness for a certain chain where a competitor has already located its facilities. In this competitive environment there also exists a hierarchical structure. It is assumed that each customer demands for two types of services for which they dedicate some buying power. Both chains' facilities are structured to be in two levels where a level-1 facility can only serve for demand type-1 and a level-2 facility can satisfy both type-1 and type-2 demands, which is called a successively inclusive service hierarchy. Here $k \in \{1, 2\}$ is the index for both service types and facility levels. We try to answer the following questions:

(i) How many level-1 facilities to open?

(ii) How many level-2 facilities to open?

(iii) Where should level-1 and level-2 facilities be located?

(iv) What should be the attractiveness of each level of facilities?

Customer choice among these facilities is modeled by using a hybrid of deterministic and probabilistic patronizing behavior. Demands (buying powers) of customers are assumed to be fixed which means a static competition is adopted. Customers are located at a discrete set of points also called potential sites, some of which will be chosen as facility locations. There is an associated fixed cost f_k which represents the fixed cost of opening a level-k facility. Since the location space is discrete we define the binary decision variable x_{ik} as

$$x_{ik} = \begin{cases} 1 & \text{if a level-}k\text{ facility is located at potential site } i \\ 0 & \text{otherwise} \end{cases}$$

Because of our hierarchical structure, namely successively inclusive service hierarchy, we do not expect that in the optimal solution a level-1 facility and a level-2 facility are both opened at the same potential site. Since level-2 facilities can satisfy both type-1 and type-2 demands and a level-2 facility is more attractive than a level-1 facility, opening a level-1 facility at the same site of a level-2 facility will bring only negative effect on the objective value. When systems are nested, the constraint that is explained below is a widely used method for optimality (Narula and Ogbu, 1979)

$$\sum_{k=1}^{2} x_{ik} \leq 1 \quad i = 1, ..., n \tag{3.1}$$

This constraint guarantees that at most one facility (that can be level-1 or level-2) is located at each site.

Each level of facility has an unknown attractiveness. Let A_{ik} is the attractiveness of a level-k facility that is located at potential site i and is assumed to be continuous. ua_k is the unit cost of attractiveness of a level-k facility. Clearly it is seen that both fixed costs and unit attractiveness costs are independent from the potential site where the facility is located. In other words, any cost that exist in the model is not affected by the location of the facility. We have to emphasize that all type-1 facilities have the same attractiveness value likewise all type-2 facilities have the same attractiveness value. Moreover, a higher level facility should certainly have a higher attractiveness value than a lower level facility since in real life situations hierarchical competitive chains generally follow this strategy. To provide these assumptions we employ the constraints:

$$\sum_{h=k}^{2} x_{ih} A_h = A_{ik} \quad i = 1, ..., n; k = 1, 2 \tag{3.2}$$

$$A_2 \geq A_1 \tag{3.3}$$

where the variable A_k, $k = 1, 2$ is used to force all facilities of the same level to have the same attractiveness values. Since we have an optimality constraint for facility location which was mentioned above, the attractiveness of a level-k facility that is located at site i namely A_{ik}, is guaranteed to be equal to A_k which is valid for all level-k facilities. Furthermore, if there is no facility at site i, then the attractiveness of that site will be zero. However if we do not have constraint (3.1), constraint (3.2) would be wrong. Constraint (3.3) ensures that attractiveness of a level-2 facility is higher than attractiveness of a level-1 facility. We have to underline that while determining the attractiveness values of competitor's facility, the same rule is employed ($CA_2 \geq CA_1$).

The objective is to maximize the total net profit of one's own chain. Total net profit is obtained by subtracting the total fixed costs and attractiveness costs of located facilities from the total captured market share. Before explaining the objective function in detail we would like to give information about customer choice. Note that customer j has different buying powers for each type of services: BP_{jk} is the money that is dedicated to type-k service by customer j. For consistency please always assume that demand and buying power, both have the same meaning.

As mentioned earlier patronizing behavior is a hybrid of deterministic and probabilistic behavior. Firstly the deterministic part takes place. A type-k demand of a customer j, BP_{jk} is allocated to both one's own chain's closest level-k or higher level facility that is located at site i, and competitor's closest level-k or higher level facility c. We have two levels of facilities and two types of demand. Note that the closest facility to a type-1 customer is determined from the set of all level-1 and level-2 facilities whereas for type-2 demand the closest facility is obtained from the set of only level-2 facilities. For type-1 demand, customer can be satisfied by a level-1 facility or by a level-2 facility. It means that a type-1 customer can be served by a level-2 facility as long as that facility is the closest one to the customer. However to satisfy type-2 demand, a customer should surely be allocated to a level-2 facility. To illustrate the idea, defining the allocation variables will be beneficial:

$$b_{cjk} = \begin{cases} 1 & \text{if type-}k\text{ demand of customer } j \text{ is allocated to competitor's facility } c. \\ 0 & \text{otherwise.} \end{cases}$$

An allocation of BP_{jk} is done if competitor's level-k (or higher level) facility c is the closest facility for customer j among competitor's all level-k or higher level facilities. Likewise

$$y_{ijk} = \begin{cases} 1 & \text{if type-}k\text{ demand of customer } j \text{ is allocated to one's own chain's facility located at site } i. \\ 0 & \text{otherwise.} \end{cases}$$

Here allocation is done if one's own chain's level-k or higher level facility located at site i is the closest facility for customer j among one's own chain's all level-k or higher level facilities. Since the competitor has existing facilities, all b_{cjk} values are determined by applying the proximity rule. The allocation of customers to one's own chain facilities is determined by binary variables y_{ijk} which relate to location variables x_{ik}.

$$\sum_{h=k}^{2} x_{ih} \geq y_{ijk} \quad i = 1, ..., n; k = 1, 2; j = 1, ..., m \tag{3.4}$$

Equation (3.4) simply states that customer j can get type-k service from site i only if a level-k or a higher level facility is located at i. Hence, y_{ijk} can take value 1 only if a level-k or higher level facility is located at site i. To ensure that this facility is the closest facility for customer j we use the constraint below:

$$\sum_{i=1}^{n} d_{ij} y_{ijk} = d_{tj} + M(1 - \sum_{g=k}^{2} x_{tg}) \quad t = 1, ..., n; j = 1, \ldots, m; k = 1, 2 \tag{3.5}$$

which imposes that type-k demand of customer j is allocated to the closest level-k or higher level facility where M is a sufficiently big number. In other words if the closest facility is located at site i, y_{ijk} is set to one by that constraint.

We have to mention that here allocation of a customer to a facility does not mean that this customer will patronize that facility. In other words, customer j will not spend the whole BP_{jk} at the facility located at site i. Both demand types of all customers are ensured to be allocated to two facilities, one of them belongs to competitor and one of them belongs to one's own chain. This is done to ensure that customer j will divide his/her buying power among these two facilities, which means that BP_{jk} will be shared among competitor's facility c and one's own chain's facility located at site i. We have to mention that this is a kind of single source assumption and is realized by using the constraint:

$$\sum_{i=1}^{n} y_{ijk} = 1 \quad j = 1, ..., m; k = 1, 2 \tag{3.6}$$

We define the probability or frequency of shopping of customer j for demand type-k, after he/she is allocated to one's own chain's closest facility located at site i and competitor's closest facility c as:

$$P_{ijk} = \frac{A_{ik}/(d_{ij}+1)^{\lambda}}{A_{ik}/(d_{ij}+1)^{\lambda} + CA_{ck}/(dc_{cj}+1)^{\lambda}} \tag{3.7}$$

where

P_{ijk} = the probability (frequency) that customer j will patronize facility located at site i for demand type-k,

A_{ik} = attractiveness of a level-k facility located at site i,

CA_{ck} = attractiveness of a level-k facility c that belongs to competitor,

d_{ij} = between customer j and facility that is located at site i,

cd_{cj} = Euclidean distance between customer j and competitor's facility c,

λ = sensitivity parameter for distance.

It is seen that a customer splits his/her demand between two facilities proportional to the attractiveness of a facility and inversely proportional to some power of the distance, which is a version of Huff's gravity based model. Here, attractiveness can be assumed to be either a single attribute or a combination of attributes.

In the literature many papers modify Huff's distance function. Most of them adapt only the distance between a customer and facility. The motivation in using a distance correction factor in Drezner and Drezner (1997) is that customers are not located at discrete points in reality but they are distributed continuously in the plane. To see the effect of using discrete customer locations, they develop a continuous customer location model and compare it with the same model but which has discrete customer locations. They observe that significant changes occur in the optimal locations. However using continuous customer locations requires solving a double integral and it could be difficult in some cases. As a result, they formulate a distance correction approach in a discrete model which is represented as: $[d^2 + 0.16scir]^{\frac{\lambda}{2}}$ where

d = distance between the center of the circle in which customers are located and the facility,

$scir$ = the area of the circle($scir = \pi r^2$),

λ = parameter for sensitivity of distance.

In many of the discrete space models another approach is adopted. $(d_{ij} + 1)^{\lambda}$ is used as the distance function. It is claimed that d_{ij}^{λ} as the denominator may cause undesired situations. If a facility is located on customers or sufficiently close to customers, then the attractiveness of the facility will be unimportant and that particular facility will capture almost all buying power of that customers. We have to emphasize that in Huff-based models unless a correction factor or an additional term is used, as distance goes to zero the term $\frac{A_{ik}}{(d_{ij})^\lambda}$ goes to infinity. Moreover in the solution step especially in discrete space problems when the sites where facilities are located are also locations of customers, many division by zero issues appear (Drezner and Drezner, 2007). Aboolian et al. (2007b) who also use $(d_{ij}+1)^\lambda$ in a discrete space Huff-like model show that $u_{ij} = A_j \exp(-\lambda d_{ij})$ is completely equivalent to $u_{ij} = A_j(d_{ij}+1)^\lambda$. As a result of the reasons explained above, we also utilize that kind of distance factor.

Now we can combine the deterministic and probabilistic parts of the patronizing behavior.

$$Pf_{tjk} = \frac{A_{tk}y_{tjk}/(d_{tj}+1)^\lambda}{\sum_{i=1}^{n} A_{ik}y_{ijk}/(d_{ij}+1)^\lambda + \sum_{c=1}^{comp} CA_{ck}b_{cjk}/(cd_{cj}+1)^\lambda} \quad t=1,..,n; j=1,..,m; k=1,2 \tag{3.8}$$

where

Pf_{tjk}	=	probability (frequency) that customer j shops for type-k demand from potential site t,
A_{tk}	=	attractiveness of a level-k facility that is located at site t,
y_{tjk}	=	allocation variable of a type-k demand of a customer for potential site t,
d_{tj}	=	distance between customer j and site t.

If a level-k or higher level facility located at site t is not the closest level-k or higher level facility among one's own chain to customer j or if there is no level-k or higher level facility at site t, then the allocation variable y_{tjk} will be zero. This will cause Pf_{tjk} to be zero, which means that customer j will not shop from potential site t for his/her type-k demand. Actually this probability function will be greater than zero for just one potential site at which the closest level-k or higher level facility is located. Hence if we sum up the probabilities of all potential sites we will achieve the probability or frequency that a customer j shops for type-k demand from one's own chain, which is represented as:

$$P_{jk} = \frac{\sum_{i=1}^{n} A_{ik} y_{ijk}/(d_{ij}+1)^{\lambda}}{\sum_{i=1}^{n} A_{ik} y_{ijk}/(d_{ij}+1)^{\lambda} + \sum_{c=1}^{comp} CA_{ck} b_{cjk}/(cd_{cj}+1)^{\lambda}} \quad j = 1, ..., m; k = 1, 2 \quad (3.9)$$

And finally the total market share captured by one's own chain ($TCMS$]) is obtained by multiplying this probability by each customer's type-k buying power and summing the multiplication over the demand type and over the customers.

$$TCMS = \sum_{j=1}^{m}\sum_{k=1}^{2} BP_{jk} \frac{\sum_{i=1}^{n} A_{ik} y_{ijk}/(d_{ij}+1)^{\lambda}}{\sum_{i=1}^{n} A_{ik} y_{ijk}/(d_{ij}+1)^{\lambda} + \sum_{c=1}^{comp} CA_{ck} b_{cjk}/(cd_{cj}+1)^{\lambda}} \quad (3.10)$$

$$= \sum_{j=1}^{m}\sum_{k=1}^{2} BP_{jk} P_{jk}$$

The total captured market share by one's own chain is the first part of our objective function. In median based models where the number of facilities to be opened is predetermined, the objective is usually maximizing the market share. However in our model, the number of each type of facilities to be opened is not predetermined and is also a decision variable. The total fixed costs of locating facilities can be written as

$$TFC = \sum_{i=1}^{n}\sum_{k=1}^{2} f_k x_{ik} \quad (3.11)$$

We have to point out that a chain may consist of a mixture of level-1 and level-2 facilities, or only level-1 facilities, or only level-2 facilities. It is also possible that the owner of the chain may prefer not to open any facilities in the presence of a negative profit. The total attractiveness cost of the chain is given as

$$TCA = \sum_{i=1}^{n} \sum_{k=1}^{2} ua_k A_k x_{ik} \tag{3.12}$$

Finally the total net profit is

$$TNP = \sum_{j=1}^{m} \sum_{k=1}^{2} BP_{jk} \frac{\sum_{i=1}^{n} A_{ik} y_{ijk} / (d_{ij}+1)^{\lambda}}{\sum_{i=1}^{n} A_{ik} y_{ijk} / (d_{ij}+1)^{\lambda} + \sum_{c=1}^{comp} CA_{ck} b_{cjk} / (cd_{cj}+1)^{\lambda}} - \sum_{i=1}^{n} \sum_{k=1}^{2} f_k x_{ik} \tag{3.13}$$

$$- \sum_{i=1}^{n} \sum_{k=1}^{2} ua_k A_k x_{ik}$$

The total net profit is obtained by subtracting the costs of forming a chain (which includes fixed costs of facilities and attractiveness costs) from the total captured market share. Obviously we can argue that total captured market share is directly proportional to the number of facilities to be opened since the distances will be smaller when there are more facilities in the area. Hence the first part of the model ($TCMS$) will always attempt to open more facilities with high attractiveness values, however the second part of the model will cause a decrease in the objective function as more facilities are opened (TFC) and as facilities have high attractiveness values (TCA).

In the model there is an interplay between number of facilities to be opened and attractiveness values. One can try to find the optimal solution by opening less facilities with high attractiveness values -which will cause high distances between facilities and customers- or reversely by opening many facilities each having less attractiveness values in which distances between facilities and customers will be significantly small. Also trade-off situations between the number of level-1 and the number of level-2 facilities to be opened, included by the model.

We name this model Competitive and Hierarchical Location Problem(COHILP) and the complete formulation becomes the following mixed integer nonlinear program:

i	=	index of potential sites	$i = 1, ..., n$
j	=	index of customers	$j = 1, ..., m$
k	=	index of both demand types and facility levels	$k = 1, 2$
c	=	index of competitor's facilities	$c = 1, ..., comp$

Parameters:

BP_{jk}	=	buying power of customer j for demand type-k
CA_{ck}	=	attractiveness value of a level-k facility c
CA_k	=	attractiveness value of a level-k facility that belongs to competitor
b_{cjk}	=	allocation parameter of a type-k demand of customer j to facility c
cd_{cj}	=	distance between competitor's facility c and customer j
d_{ij}	=	distance between potential site i and customer j
λ	=	parameter for sensitivity of distance
f_k	=	fixed cost of opening a level-k facility
ua_k	=	unit cost of attractiveness of a level-k facility
M	=	a sufficiently big number

Variables: $x_{ik} = \begin{cases} 1 & \text{if a level-}k\text{ facility is located at potential site } i \\ 0 & \text{otherwise} \end{cases}$

$$y_{ijk} = \begin{cases} 1 & \text{type-}k\text{ demand of customer } j \text{ is allocated to one's own chain's facility located at site } i. \\ 0 & \text{otherwise} \end{cases}$$

A_{ik} = attractiveness value of a level-k facility located at site i

A_k = common attractiveness value of all level-k facilities

$$\text{COHILP:}\quad \max \sum_{j=1}^{m}\sum_{k=1}^{2} BP_{jk}\frac{\sum_{i=1}^{n} A_{ik}y_{ijk}/(d_{ij}+1)^{\lambda}}{\sum_{i=1}^{n} A_{ik}y_{ijk}/(d_{ij}+1)^{\lambda} + \sum_{c=1}^{comp} CA_{ck}b_{cjk}/(cd_{cj}+1)^{\lambda}} - \sum_{i=1}^{n}\sum_{k=1}^{2} f_k x_{ik} - \sum_{i=1}^{n}\sum_{k=1}^{2} ua_k A_k x_{ik} \tag{3.14}$$

subject to

$$\sum_{h=k}^{2} x_{ih} \geq y_{ijk} \qquad i=1,...,n; j=1,...,m; k=1,2 \tag{3.15}$$

$$\sum_{i=1}^{n} y_{ijk} = 1 \qquad j=1,...,m; k=1,2 \tag{3.16}$$

$$\sum_{k=1}^{2} x_{ik} \leq 1 \qquad i=1,...,n \tag{3.17}$$

$$\sum_{i=1}^{n} d_{ij}y_{ijk} = d_{tj} + M(1 - \sum_{g=k}^{2} x_{tg}) \qquad t=1,...,n; j=1,\ldots,m; k=1,2 \tag{3.18}$$

$$\sum_{h=k}^{2} x_{ih}A_h = A_{ik} \qquad i=1,...,n; k=1,2 \tag{3.19}$$

$$A_2 \geq A_1 \tag{3.20}$$

$$A_k \geq 0 \qquad k=1,2 \tag{3.21}$$

$$x_{ik} \in \{0,1\} \qquad i=1,\ldots,n; k=1,2 \tag{3.22}$$

$$y_{ijk} \in \{0,1\} \qquad i=1,\ldots,n; j=1,\ldots,m; k=1,2 \tag{3.23}$$

4. Solution Procedure

Since the problem is a nonlinear-mixed integer programming model using exact solution techniques do not guarantee to find the global optimal solution moreover trying to calculate the objective values of all feasible solutions will cause a great computational effort. SBB solver within GAMS v22.0 is employed for different instances. With fixed parameters the results obtained by SBB solver are not as good as expected. Heuristic methods may turn out to be efficient for solving combinatorial optimization problems. Hence SA algorithm which includes add, drop and swap moves, are employed in the outer loop to determine the number of facilities to be opened and their locations. For finding the continuous attractiveness values Nelder-Mead Simplex search and Fibonacci search algorithms are used in the inner loop after level-1 and level-2 facilities are located. The main motivation of using SA is the non-convexity of the objective function. If a function is not convex or concave, then many local optimal solutions exist and using a steepest ascent algorithm may cause stucking at a local optimal solution. To prevent this, SA algorithm is a widely used method which moves to better solutions with probability 1 and to worse solutions with some probability which is proportional to the quality of the solution. In the literature the most preferred meta-heuristic algorithms for facility location problems are SA and TS.

We will first mention the general procedure of the algorithm and then explain the operators and the algorithms used in the inner loop.

4.1. Initial Solution

We obtain the initial solution by opening a level-1 facility to a randomly selected site and then by opening a level-2 facility to another randomly selected site which is different from the location of level-1 facility.

Like Weizsfeld algorithm, the final solution of our algorithm strongly depends on the initial solution hence we run our algorithm multiple times.

4.2. Simulated Annealing

After the initialization step, an initial temperature is determined and the initial solution is sent to the improvement loop. The following operations are performed for a predetermined number of iterations:

Three operators are used: (i) Add level-1: It opens a level-1 facility at a site (ii) Add level-2: It opens a level-2 facility at a site. (iii) Drop level-1: It closes a level-1 facility that was located. (iv) Drop level-2: It closes a level-2 facility that was located (v) Swap: It consists of three steps which are swap a level-1 facility, swap a level-2 facility and exchange the locations of a level-1 and a level-2 facility. We select a random move. After determining a new solution with respect to number and locations of the facilities, this new solution is sent to Simplex search or Fibonacci search algorithm to find the attractiveness values depending on the number of variables. For instance, if the number of open level-1 facilities and the number of open level-2 facilities are both positive, then it is clear that we have to find two attractiveness values for each type. Hence a simplex search is employed. On the other hand, if open facilities are all of the same level, then we only need the attractiveness value of a single level and we employ Fibonacci search. Finally, the objective value of the new solution is calculated with respect to locations of the facilities and attractiveness values. SA moves to a new solution with probability 1 if its objective value is greater than the objective value of the existing solution or with probability in (0,1) if the objective value of the new solution is worse than the objective value of the existing solution. This probability is calculated using the temperature and the objective values of both existing and new solution. If the difference between objective values is small, then the probability of accepting the new solution will be high, which means that as the quality of the new solution increases the probability of accepting the solution increases. At the end of the improvement loop the temperature is updated by multiplying with a coefficient between 0 and 1, which is called "cooling" step. Note that the probability of accepting a worse solution is directly proportional to the temperature hence at the beginning, the algorithm accepts worse results with a high probability which provides a better exploration of the solution space. At each increment of the improvement loop the value of the temperature decreases implying that the probability of accepting bad solutions decreases at each step.

4.3. Termination Criterion

As mentioned earlier, during the improvement loop a predetermined number of iterations are made. And the best objective value among these iterations is determined. We compare the best objective value of the current improvement loop with the best objective value found in the previous improvement loop and if there is no improvement we increment the *no_improvement* parameter by one, otherwise the value of *no_improvement* is set to one. The algorithm stops if there are no improvements for consecutive *max_no_imp* loops. In other words, when the value of *no_improvement* is equal to *max_no_imp* which is a certain number determined at the beginning of the algorithm, SA is terminated.

Notation

initial	=	number of different initial solutions
max *_initial*	=	maximum number of different initial solutions
no_improvement	=	number of non improvements
max_no_imp	=	maximum number of non improvements
intem	=	initial temperature

4.4. Neighborhood Structure

We two strategies each having different add-drop rules within the SA algorithm. The first strategy, namely SA with Second Closest Add Drop (SASCAD), uses add-by-second-closest and drop-by-second-closest rules. The second strategy which is called SA with Random Add Drop (SARAD), includes random add-drop rules which will be defined below (Brimberg et al 2000).

4.4.1. Add Move

Add moves are employed to open new facilities on the potential sites where there is no facility that is already located. Both add moves and drop moves help us to determine the number of level-1 and level-2 facilities to be opened and the locations of these facilities.

4.4.1.1. Add By Second Closest(ASC). Let s be the objective value of the current solution and r_u be the objective value of the solution when a facility is temporarily added to an unoccupied site u. Assume that US is the set of all unoccupied sites and $u \in US$. Note that when a facility is located at u, then a reallocation of the customers who are now closer to the newly opened facility than their current facility (which becomes the second closest) is made. After the reallocation step the objective value r_u is calculated and the difference in the objective value is then equal to $r_u - s$. This procedure is repeated for all unoccupied sites and a facility is located at the site $u_{\max} = \max_u \{(r_u - s)\,,\ u \in US\}$.

4.4.1.2. Random Add(RA). Randomly choose an unoccupied site u where $u \in US$ and locate a facility in the site u.

4.4.2. Drop move

Drop moves are employed to close an open facility.

4.4.2.1. Drop by Second Closest(DSC). Let $z_\emptyset$ be the objective value of the solution when a facility located at site o is temporarily deleted. And customers allocated to this facility, are reallocated to the second closest facility. Assume that OS is the set of sites where there are facilities already located and $o \in OS$. After the reallocation step the objective value $z_\emptyset$ is calculated and the difference in the objective value is now equal to $s - z_\emptyset$. This procedure is repeated for all occupied sites and the facility which is located at $o_{\min} = \min_o \{(s - z_\emptyset), o \in OS\}$ is dropped.

We have to emphasize that generally an increase is expected in the add-by-second-closest move and a decrease is expected in the drop-by-second-closest-rule. However in our case there are costs in the objective function, and opening a facility may have negative effect if the total costs of opening that facility is greater than the contribution (captured market share) of that facility. Also dropping a facility may have a positive effect because of the same reason. This situations may appear if a facility is located at a site which is too far away from customers or in the presence of market cannibalization.

4.4.2.2. Random Drop(RD). Randomly choose an already located facility at site o where $o \epsilon OS$ and drop that facility.

4.4.3. Swap Moves

If a p-median version of the problem is desired to be solved then employing only the swap moves in the simulated algorithm will be beneficial. It consists of 3 steps:

1) Swap a level-1 facility: Randomly choose a level-1 facility located at site o from the set OS. Also choose an unoccupied site u randomly from the set US. Move the level-1 facility from o to u.

2) Swap a level-2 facility: Randomly choose a level-2 facility located at site o from the set OS. Also choose an unoccupied site u randomly from the set US. Move the level-2 facility from o to u.

3)Exchange: Choose an existing level-1 facility, and level-2 facility randomly, then exchange their locations.

4.5. Simplex Search Algorithm

When both types of facilities are opened in the outer loop, then the optimization of two continuous decision variables are required: A_1 the attractiveness variable of level-1 facilities, and A_2 the attractiveness variable of level-2 facilities.

If one has to find optimal values of more than one continuous variables, then simplex search used for unconstrained optimization is an appropriate algorithm. It was first proposed by Nelder and Mead (1965). However we employ a revised simplex search algorithm as described by Humphrey and Wilson (2000).

Since we have two unknown variables, the simplex is a triangle. Initially three vertices of the triangle should be determined. It can be done randomly or using some criteria. For our problem the first coordinates of the vertices, which represent the attractiveness variable of a level-2 facility (A_2), are chosen randomly from the interval $[0, \omega CA_2]$ where w is a predetermined coefficient and CA_2 is the level-2 attractiveness value of competitor. Level-1 attractiveness values are chosen randomly from the interval $[0, A_2]$. We have to underline that these intervals are determined only for initialization and it does not mean that the final values of attractiveness of each level will be in this interval.

The algorithm starts by calculating the objective values for each vertex (in the outer loop locations of the facilities are already found) and the simplex is sorted from the best value (largest) to worst value (smallest). The main idea is then changing the worst vertex to a better one by using some procedures namely: reflection, expansion, contraction.

The algorithm is repeated until one of two stopping conditions are satisfied: (i) The difference between the edges in the simplex are smaller than a predetermined tolerance number. (ii) There is not a convergence but the number of iterations made exceeds a certain limit. In this case, a local maximum cannot be found.

4.6. Fibonacci Search Algorithm

If only one type of facilities are opened in the outer loop (assume level-1 facilities are opened), then we only need to find the optimal value of A_1 since A_2 will definitely be zero. In other words, an optimization of one variable is required.

Fibonacci search is an appropriate algorithm for our purpose. It is a line search procedure which has the aim of maximizing a function θ over a closed bounded interval ($[\psi_q, \sigma_q]$ for iteration q). Only in the first iteration two functional evaluations are made and then at each iteration one evaluation is performed. The reduction of the interval varies from one iteration to another. The algorithm is based on Fibonacci sequence $\{F_v\}$ where

$$\begin{aligned} F_{v+1} &= F_v + F_{v-1} \quad v = 1, 2, ... \\ F_0 &= F_1 = 1 \end{aligned} \tag{4.1}$$

Unlike other algorithms, Fibonacci search procedure requires the total number of observations to be predetermined. This is because at each reduction of the interval, Fibonacci numbers, which depend on the total number of observations (ϖ) are used. Hence ϖ should be chosen such that $F_{\varpi} > (\sigma_1 - \psi_1)/\zeta$ where ζ is the final length of the interval. We have to emphasize that as an initial interval $\psi_1 = 0$ and $\sigma_1 = \Omega CA_2$ is assumed where Ω is a predetermined coefficient (Bazaraa et al., 2005).

Both Fibonacci search and simplex search algorithms are used for unconstrained optimization. However for attractiveness variables we have constraints (3.20) and (3.21). For the non-negativity constraint the Fibonacci search algorithm is not an issue since the interval from where optimal solution is found, is predetermined. So fixing the left side of the interval by zero overcomes the problem. However for the Simplex search algorithm, a modification is handled to find non-negative values of attractiveness. If the modification is not made, it is possible to find negative attractiveness values because in the objective function there exists total attractiveness costs of opening a chain and this term is subtracted from the total captured market share. If the attractiveness values are negative, the total cost of opening a chain will bring a positive contribution to the objective value while the total captured market share will be negative. Moreover, if the absolute value of the negative effect of the total captured market share is less than the positive effect of total unit attractiveness costs, then the algorithm will assign negative values to attractiveness. Hence a modification is required.

Constraint (3.20) can be satisfied by using appropriate fixed costs and unit attractiveness costs. Therefore, no modification is made for this constraint. Only costs are chosen carefully. Even if a solution in which $A_1 \geq A_2$ was obtained, it would just be infeasible and would not be taken into account.

5. Computational Results

In the following sections, we present the results of our computational experiments performed by using two different strategies: SASCAD and SARAD.

5.1. Experimental Setup

Since the model is sensitive to the parameters, several test problems have been generated. The parameters are investigated in two levels and all combinations of different levels are taken into account.

The problem size is handled in three levels:

(i) 20-customer problem

(ii) 50-customer problem

(iii) 100-customer problem

As mentioned earlier, customer sites are also potential sites for opening facilities. Hence in a 20-customer problem $n = m = 20$, in 50-customer problem $n = m = 50$ and in 100-customer problem $n = m = 100$. The coordinates of the potential sites which are also customer sites are generated randomly. The customers and the sites are located in the two-dimensional Euclidean plane. The buying power of each customer BP_{jk} is randomly generated from the interval $[1, 50]$ hence buying powers are uniformly distributed between 1 and 50.

For each problem size, 32 problems are created in which the following parameters all have two levels: Number of level-1 facilities that belong to competitor ($NL1$), number of level-2 facilities that belong to competitor ($NL2$), locations of competitor's facilities (Comp.'s Loc.), fixed costs of opening facilities (f_k), unit attractiveness cost

(ua_k) of each level of facilities. For a better understanding, the tables below are used to illustrate the design of experiments. Here $S \in \{20, 50, 100\}$ where 20 represents 20-customer problem,etc.

Table 5.1. Experimental Design for $NL1 = 2, NL2 = 2$

Prob. No	**Comp.'s Loc.**	f_k	ua_k
S.1	1	L	L
S.2	1	L	H
S.3	1	H	L
S.4	1	H	H
S.5	2	L	L
S.6	2	L	H
S.7	2	H	L
S.8	2	H	H

Table 5.2. Experimental Design for $NL1 = 2, NL2 = 1$

Prob. No	**Comp.'s Loc.**	f_k	ua_k
S.9	1	L	L
S.10	1	L	H
S.11	1	H	L
S.12	1	H	H
S.13	2	L	L
S.14	2	L	H
S.15	2	H	L
S.16	2	H	H

The locations of competitor's facilities are chosen randomly from the potential sites. However the locations of competitor's facilities have a strong effect on the optimal solution. Hence for the first four problem locations of competitors' facilities are fixed. 1 represents the first locations of facilities. We change the locations of competitors' facilities for the second four problems. 2 represents the second locations of the facilities. The fixed costs of opening facilities are determined by trying different values and observing the behavior of the model. Here L means "low" and H means "high ". Unit attractiveness costs are also obtained by trial.

Table 5.3. Experimental Design for $NL1 = 1, NL2 = 2$

Prob. No	**Comp.'s Loc.**	f_k	ua_k
S.17	1	L	L
S.18	1	L	H
S.19	1	H	L
S.20	1	H	H
S.21	2	L	L
S.22	2	L	H
S.23	2	H	L
S.24	2	H	H

Table 5.4. Experimental Design for $NL1 = 1, NL2 = 1$

Prob. No	**Comp.'s Loc.**	f_k	ua_k
S.25	1	L	L
S.26	1	L	H
S.27	1	H	L
S.28	1	H	H
S.29	2	L	L
S.30	2	L	H
S.31	2	H	L
S.32	2	H	H

5.2. Comparison of SASCAD and SARAD

Now we would like to declare the values of the parameters that are used in the algorithm before comparing the results of them.

Table 5.5. Parameters in Simulated Annealing Algorithm

$\max\_no\_imp$	$intem$	$coef$	$impro\_crt$	M
5	500	0.7	0.05	1000000

Table 5.6. Parameters in Simplex Search Algorithm

α	β	γ	ρ	w
1	0.5	2	0.01	8

Table 5.7. Parameters in Fibonacci Search Algorithm

ζ	ϵ	q
0.005	0.1	8

Table 5.8. Parameters in the model

λ	CA_1	CA_2
2	10	20

Deciding on the initial temperature is a challenge. Normally in SA algorithm, the initial temperature is determined by choosing an initial probability and finding Δ_{avg} as described below:

$$\Delta_a = \Psi_a - \Psi'_a \qquad a = 1, 2, ...v \tag{5.1}$$

where

Ψ_a = objective value of existing solution,

Ψ'_a = objective value of new solution,

v = total number of observations,

Δ_a = difference between the objective values of new and existing solutions,

Δ_{avg} = average difference between objective values of new and existing solutions.

$$\Delta_{avg} = \frac{\Delta_a}{v} \tag{5.2}$$

The probability of accepting worse solutions:

$$prb = e^{-\Delta_{avg}/intem} \tag{5.3}$$

Once the probability is chosen, then the initial temperature is calculated as:

$$intem = \frac{\Delta_{avg}}{\ln(1/prb)} \tag{5.4}$$

The new solution is found by neighborhood search from the existing solution. For our algorithm these are add, drop and swap operators. However, we do not employ this procedure while deciding on the initial temperature. Since the algorithm includes add and drop operators, values of Δ_a's will not be close to each other. The reason of this is the cannibalization effect. At the beginning when there is no facility opened, opening a facility will bring a large contribution to the objective value whereas when there are a lot of facilities already existing, opening a facility will bring a small contribution. It may even be expected that opening a facility may cause a negative effect on the objective value. Hence finding an average difference will not be meaningful since the deviation will be large. The initial temperature is chosen by trial and error approach.

Since we mainly focus on hierarchical gross markets, it is appropriate to take the value of λ as 2. In the literature it is advised to choose a value from the interval $[2, 2.5]$ for grocery stores (Bell et al, 1998). The parameters given above are fixed ones for all problem sizes. The rest of the parameters change for each problem size $(f_k, ua_k, \max\_iter, \max\_initial)$.

We would like to mention that the costs have an effect on the optimal solutions. If too high fixed costs or unit attractiveness costs were set, then the model would try to open as less facilities as it could with low attractiveness values. It would even choose not to open any facilities if the total cost of constructing a chain exceeds the total captured market share. If the costs were very low, then the model would tend to open too many facilities with very high attractiveness values. Hence we try to choose the values of costs such that the number of facilities and their attractiveness values can be realistic.

In the event of having low costs, the number of possible solutions would be extremely high. Too many trade-off scenarios may appear. For instance, it may be chosen to open too many facilities with low attractiveness values in order to minimize the distances between facilities and customers. Or it may be preferred to open only a few facilities with high attractiveness values. Furthermore deciding on the number of each level of facilities to be opened would be challenging hence the probability of finding the optimal solution would dramatically decrease. In fact the most difficult part of the solution procedure is deciding on the number of each level of facilities to be opened.

5.2.1. 20-customer Problem

Here we compare SASCAD with SARAD in 20-customer problem. The parameters that change for each problem size are:

Table 5.9. Parameters for 20-customer problems

$\max\_iter$	$\max\_initial$
50	9

for 20-customer problem. And the costs are:

The maximum objective values (TNP^*), and CPU times which are obtained by employing two strategies are:

Table 5.10. Costs for 20-customer problems

	f_1	f_2	ua_1	ua_2
L	20	60	1	3
H	50	150	3	9

Table 5.11. Results and Comparison of SARAD and SASCAD for $NL1 = 2$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	%$impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
20.1	361.05	361.05	100.00	41	114
20.2	180.65	192.06	94.06	71	148
20.3	254.62	254.62	100.00	33	109
20.4	55.68	55.68	100.00	58	136
20.5	358.17	422.55	84.76	61	129
20.6	210.56	252.05	83.54	57	171
20.7	264.24	328.00	80.56	44	137
20.8	90.06	132.36	68.04	44	165

Percent improvements obtained by SARAD is calculated as:

$$\%impr = \frac{TNP_{SARAD}}{TNP_{SASCAD}} 100 \tag{5.5}$$

It is seen that in most of the problems SARAD obtains the same results as SASCAD does while taking a relatively small CPU time.

It is observed that for the problem 20.18 SARAD results even with a better objective value that SASCAD. However in a few problems SASCAD gives better results than SARAD but the difference is negligible.

Table 5.12. Results and Comparison of SARAD and SASCAD for $NL1 = 2$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
20.9	551.43	551.43	100.00	43	116
20.10	409.21	423.91	96.53	58	135
20.11	453.64	453.64	100.00	40	129
20.12	276.57	276.57	100.00	98	138
20.13	461.94	461.94	100.00	54	139
20.14	263.70	272.89	96.63	59	157
20.15	359.86	359.86	100.00	50	140
20.16	160.87	160.87	100.00	80	157

Table 5.13. Results and Comparison of SARAD and SASCAD for $NL1 = 1$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
20.17	486.11	499.84	97.25	57	122
20.18	372.38	371.96	100.11	96	162
20.19	384.89	384.89	100.00	59	134
20.20	230.13	230.13	100.00	106	172
20.21	386.64	386.63	100.00	44	120
20.22	226.43	226.43	100.00	62	173
20.23	286.82	286.82	100.00	44	133
20.24	102.90	175.68	58.57	65	151

For all problems SARAD is faster than SASCAD. In 21 problems the objective values of SARAD overlap with the objective values of SASCAD and for one problem SARAD is better than SASCAD.

Table 5.14. Results and Comparison of SARAD and SASCAD for $NL1 = 1$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
20.25	604.33	604.33	100.00	48	124
20.26	464.62	464.62	100.00	63	148
20.27	514.33	514.33	100.00	57	123
20.28	361.86	361.86	100.00	91	154
20.29	476.64	486.91	97.89	44	115
20.30	296.23	296.23	100.00	73	178
20.31	385.90	385.90	100.00	41	136
20.32	192.99	192.99	100.00	54	154

5.2.2. 50-customer Problem

We compare the results of SASCAD with SARAD in 50-customer problem. The values of the parameters and costs that are used in 50-customer problem are listed below. The reason of decreasing the number of maximum initial solutions is the time constraint. The run times of the programs exponentially increase as the problem size increases. The costs are also adapted to 50-customer problem because the number of customers increase from 20 to 50 which causes an increase in the total market share.

Table 5.15. Parameters for 50-customer problems

$\max\_iter$	$\max\_initial$
80	4

Table 5.16. Costs for 50-customer problems

	f_1	f_2	ua_1	ua_2
L	20	75	2	8
H	50	200	3	12

Table 5.17. Results and Comparison of SARAD and SASCAD for $NL1 = 2$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
50.1	978.75	981.62	99.71	337	2963
50.2	809.67	823.70	98.30	307	3715
50.3	853.75	853.75	100.00	207	2297
50.4	679.30	679.30	100.00	288	2853
50.5	888.58	888.58	100.00	514	2181
50.6	717.02	717.02	100.00	479	2779
50.7	763.58	763.58	100.00	246	2338
50.8	592.02	592.02	100.00	643	3019

The results are similar with the results of 20-customer problem, furthermore if one pays attention to the CPU times, it will be seen that the difference between the run times of two strategies significantly increases. In 50-customer problem SARAD obtains the same objective values with SASCAD in 21 problems out of 32. It is seen that the quality of solutions obtained by SARAD is not affected by the size of the problem. Considering the results shown in all of the tables it is concluded that SASCAD is more successful than SARAD. In fact it is an expected result since the SASCAD employs add-by-second-closest, drop-by-second-closest rules. In SASCAD, whenever a solution is sent to one of these operators, the add/drop operator tries to make an improvement in the objective value. The add operator opens a facility at an unoccupied site which has the highest contribution to the objective value. Even if each unoccupied site has a negative effect in the objective value, add-by-second-closest operator opens a facility at a site which brings the least negative effect. The same principle is also valid for the drop-by-second-closest operator. However in SARAD making an improvement is not obligated since the strategy uses random-add, random-drop rules. The aim of these operators is just searching the solution space, not improving the current solution.

Table 5.18. Results and Comparison of SARAD and SASCAD for $NL1 = 2$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
50.9	1441.30	1436.30	100.35	523	2397
50.10	1311.02	1311.02	100.00	253	2553
50.11	1316.30	1316.30	100.00	303	3727
50.12	1186.02	1186.02	100.00	262	3296
50.13	1141.79	1165.67	97.95	356	2760
50.14	988.39	1033.04	95.68	409	2601
50.15	1016.79	1068.61	95.15	580	2149
50.16	863.39	917.88	94.06	243	2662

Table 5.19. Results and Comparison of SARAD and SASCAD for $NL1 = 1$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
50.17	1257.61	1257.61	100.00	240	2467
50.18	1105.08	1105.08	100.00	390	2870
50.19	1132.61	1132.61	100.00	332	2628
50.20	980.08	980.08	100.00	286	2519
50.21	915.69	918.00	99.75	381	2199
50.22	735.88	747.47	98.45	263	2574
50.23	793.00	793.00	100.00	331	2477
50.24	622.47	622.47	100.00	217	2703

On the other hand the CPU times of SARAD are significantly less than the CPU times of the SASCAD. The factors which cause this significant difference are also the add/drop operators. Firstly we would like to investigate SASCAD. Assume we have v

Table 5.20. Results and Comparison of SARAD and SASCAD for $NL1 = 1$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	TNP^*_{SASCAD}	$\%impr$	CPU by SARAD (sec)	CPU by SASCAD (sec)
50.25	1303.19	1303.19	100.00	234	2588
50.26	1142.64	1146.56	99.66	156	2542
50.27	1178.19	1178.19	100.00	156	2559
50.28	1021.56	1021.56	100.00	203	2421
50.29	1237.64	1237.64	100.00	194	2679
50.30	1077.32	1077.41	99.99	226	2664
50.31	1112.64	1112.64	100.00	214	2012
50.32	952.41	952.41	100.00	269	2873

unoccupied sites and $(n - v)$ occupied sites. In the add-by-second-closest operator, v possible solutions are tried one by one to decide where to open a facility. In one trial after a location is selected, a Fibonacci search or simplex search algorithm is employed to find the new attractiveness values of the facilities and to calculate the objective value. This procedure is repeated for v times and the best location to open a facility is then found. The drop-by-second-closest operator also investigates each of the $(n - v)$ occupied sites for closing a facility.

However in SARAD, both the add operator and the drop operator randomly choose a site to open a facility or to close a facility. So only one Fibonacci or simplex search algorithm is applied and one objective value is calculated. We can declare that add-by-second-closest operator is almost v times slower than randomly add, and drop-by-second-closest operator is almost $(n - v)$ times slower than the randomly drop operator. We use the word "almost" because these conclusions are not strict since Fibonacci search and simplex search algorithms are both based on convergence hence CPU times of them may deviate. It is obvious that this effect will be stronger as the problem size (n) increases. As it is seen above the differences between CPU times of two strategies are negligible in 20-customer problem, whereas the differences between CPU times of SASCAD and SARAD significantly increase when the problem size is changed to 50 from 20.

5.3. Results of SASCAD

In this section you see best (max) objective values TNP^*, the number of facilities opened and the attractiveness values for the best solution, average of max_*initial* number of solutions and their standard deviations which are found by using SASCAD.

Table 5.21. Results of SASCAD for $NL1 = 2$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.1	361.05	1	1	22.96	48.18	114	351.14	17.84
20.2	192.06	2	2	3.11	6.80	148	183.68	8.91
20.3	254.62	0	1	-	59.09	109	237.64	49.16
20.4	55.68	0	1	-	10.97	136	52.41	5.35
20.5	422.55	1	1	23.40	52.38	129	405.68	32.90
20.6	252.05	1	1	9.93	8.33	171	238.41	19.47
20.7	328.00	0	1	-	60.66	137	304.22	60.79
20.8	132.36	0	1	-	11.03	165	128.65	7.37

Table 5.22. Results of SASCAD for $NL1 = 2$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.9	551.43	1	1	21.51	35.74	116	549.44	3.16
20.10	423.91	1	2	5.70	6.09	135	412.22	10.04
20.11	453.64	0	1	-	46.35	129	453.64	0.00
20.12	276.57	0	1	-	13.47	138	273.08	9.09
20.13	461.94	1	1	27.09	44.51	139	452.43	14.25
20.14	272.89	2	2	4.07	5.86	157	262.97	9.32
20.15	359.86	0	1	-	54.34	140	359.32	1.61
20.16	160.87	0	1	-	21.42	157	153.19	23.05

Table 5.23. Results of SASCAD for $NL1 = 1$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.17	499.84	1	2	6.41	16.97	122	493.19	7.45
20.18	371.96	1	2	4.99	6.17	162	355.50	21.46
20.19	384.89	0	1	-	49.35	134	381.70	9.56
20.20	230.13	0	1	-	14.43	172	229.61	1.54
20.21	386.63	1	1	27.06	46.87	120	365.70	21.50
20.22	226.43	2	1	4.53	6.71	173	227.72	2.56
20.23	286.82	0	1	-	57.78	133	264.15	68.02
20.24	175.68	0	1	-	21.57	151	90.68	45.08

Table 5.24. Results of SASCAD for $NL1 = 1$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.25	604.33	0	1	-	42.15	124	591.50	19.39
20.26	464.62	2	1	2.77	13.48	148	449.56	16.87
20.27	514.33	0	1	-	42.15	123	503.52	31.55
20.28	361.86	0	1	-	17.28	154	361.86	0.00
20.29	476.64	1	1	26.22	42.68	115	478.73	8.64
20.30	296.23	1	1	10.77	17.08	178	293.16	4.37
20.31	385.90	0	1	-	52.57	136	382.10	6.58
20.32	192.99	0	1	-	20.89	154	172.30	33.10

Table 5.25. Results of SASCAD for $NL1 = 2$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.1	981.62	1	1	23.14	43.58	2963	962.94	35.04
50.2	823.70	1	1	14.77	30.34	3715	788.46	42.55
50.3	853.75	0	1	-	52.32	2297	846.10	10.30
50.4	679.30	0	1	-	36.67	2853	632.38	59.19
50.5	888.58	0	1	-	51.57	2181	869.11	15.16
50.6	717.02	0	1	-	35.92	2779	677.15	52.56
50.7	763.58	0	1	-	51.57	2338	735.89	55.39
50.8	592.02	0	1	-	35.92	3019	587.20	8.34

Table 5.26. Results of SASCAD for $NL1 = 2$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.9	1436.30	0	1	-	39.71	2397	1423.97	14.76
50.10	1311.02	0	1	-	28.31	2553	1308.11	5.83
50.11	1316.30	0	1	-	37.88	3727	1207.10	100.23
50.12	1186.02	0	1	-	28.31	3296	1166.92	35.34
50.13	1165.67	3	1	10.28	34.80	2760	1145.93	23.58
50.14	1033.04	0	1	-	30.94	2601	1008.59	19.64
50.15	1068.61	0	1	-	45.17	2149	1068.61	0.00
50.16	917.88	0	1	-	31.74	2662	878.54	78.67

Table 5.27. Results of SASCAD for $NL1 = 1$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.17	1257.61	0	1	-	45.21	2467	1256.42	2.38
50.18	1105.08	0	1	-	30.90	2870	1078.79	52.57
50.19	1132.61	0	1	-	45.21	2628	1132.61	0.00
50.20	980.08	0	1	-	30.90	2519	974.98	10.20
50.21	918.00	0	1	-	51.13	2199	900.82	20.34
50.22	747.47	0	1	-	35.85	2574	734.09	26.77
50.23	793.00	0	1	-	51.13	2477	772.51	40.98
50.24	622.47	0	1	-	35.85	2703	587.85	34.81

Table 5.28. Results of SASCAD for $NL1 = 1$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SASCAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.25	1303.19	0	1	-	45.73	2588	1248.20	63.53
50.26	1146.56	0	1	-	33.87	2542	1143.85	3.36
50.27	1178.19	0	1	-	45.73	2559	1128.16	63.02
50.28	1021.56	0	1	-	33.87	2421	1019.83	3.46
50.29	1237.64	0	1	-	46.86	2679	1223.00	17.16
50.30	1077.41	0	1	-	32.56	2664	1070.71	13.41
50.31	1112.64	0	1	-	46.86	2012	1108.55	4.72
50.32	952.41	0	1	-	32.56	2873	947.26	8.83

5.4. Results of SARAD

Table 5.29. Results of SARAD for $NL1 = 2$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.1	361.05	1	1	22.79	47.92	41	353.26	10.42
20.2	180.65	1	2	5.39	7.62	71	160.58	30.79
20.3	254.62	0	1	-	59.09	33	254.62	0.00
20.4	55.68	0	1	-	10.97	58	55.51	0.50
20.5	358.17	1	1	26.99	55.86	61	351.22	18.26
20.6	210.56	2	1	4.26	6.93	57	199.21	16.15
20.7	264.24	0	1	-	66.20	44	264.24	0.00
20.8	90.06	0	1	-	11.39	44	90.06	0.00

Table 5.30. Results of SARAD for $NL1 = 2$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.9	551.43	1	1	22.04	35.86	43	541.19	18.32
20.10	409.21	0	2	-	7.37	58	397.28	16.29
20.11	453.64	0	1	-	46.35	40	421.76	49.93
20.12	276.57	0	1	-	13.47	98	266.04	16.29
20.13	461.94	1	1	27.02	44.39	54	423.84	42.55
20.14	263.70	1	1	11.41	17.71	59	248.52	17.03
20.15	359.86	0	1	-	54.34	50	359.32	1.61
20.16	160.87	0	1	-	21.42	80	135.18	40.08

Table 5.31. Results of SARAD for $NL1 = 1$, $NL2 = 2$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.17	486.11	1	2	9.84	17.83	57	471.88	19.46
20.18	372.38	1	2	5.57	6.17	96	355.11	19.15
20.19	384.89	0	1	-	49.35	59	384.89	0.00
20.20	230.13	0	1	-	14.43	106	227.91	4.50
20.21	386.64	1	1	26.63	47.11	44	365.95	28.60
20.22	226.43	2	1	4.61	6.66	62	205.44	20.17
20.23	286.82	0	1	-	57.78	44	266.85	47.24
20.24	102.90	0	1	-	10.49	65	99.23	12.58

Table 5.32. Results of SARAD for $NL1 = 1$, $NL2 = 1$ ($n = 20$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
20.25	604.33	0	1	-	42.15	48	601.86	4.43
20.26	464.62	2	1	2.71	13.61	63	435.53	30.39
20.27	514.33	0	1	-	42.15	57	489.83	45.41
20.28	361.86	0	1	-	17.28	91	343.67	27.21
20.29	476.64	1	1	25.90	47.39	44	464.82	23.01
20.30	296.23	1	1	11.04	17.18	73	284.17	13.31
20.31	385.90	0	1	-	52.57	41	385.05	2.53
20.32	192.99	0	1	-	20.89	54	187.92	15.19

Table 5.33. Results of SARAD for $NL1 = 2$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.1	978.75	0	1	-	52.32	337	973.32	7.64
50.2	809.67	1	1	14.03	31.51	307	801.00	7.83
50.3	853.75	0	1	-	52.32	207	845.41	11.57
50.4	679.30	0	1	-	36.67	288	658.77	24.21
50.5	888.58	0	1	-	51.57	514	879.01	11.24
50.6	717.02	0	1	-	35.92	479	707.31	13.37
50.7	763.58	0	1	-	51.57	246	761.58	4.00
50.8	592.02	0	1	-	35.92	643	549.69	60.15

Table 5.34. Results of SARAD for $NL1 = 2$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.9	1441.30	0	1	-	37.88	523	1406.49	37.24
50.10	1311.02	0	1	-	28.31	253	1308.06	3.65
50.11	1316.30	0	1	-	37.88	303	1304.56	23.48
50.12	1186.02	0	1	-	28.31	262	1184.15	3.74
50.13	1141.79	0	1	-	47.08	356	1141.49	0.34
50.14	988.39	0	1	-	32.73	409	940.98	42.37
50.15	1016.79	0	1	-	47.08	580	1016.79	0.00
50.16	863.39	0	1	-	32.73	243	847.69	22.23

Table 5.35. Results of SARAD for $NL1 = 1$, $NL2 = 2$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.17	1257.61	0	1	-	45.21	240	1182.74	82.76
50.18	1105.08	0	1	-	30.90	390	1079.22	30.77
50.19	1132.61	0	1	-	45.21	332	1127.52	3.81
50.20	980.08	0	1	-	30.90	286	956.33	43.38
50.21	915.69	0	1	-	53.77	381	833.46	112.75
50.22	735.88	0	1	-	37.85	263	702.13	48.40
50.23	793.00	0	1	-	51.13	331	789.10	6.35
50.24	622.47	0	1	-	35.85	217	618.17	5.53

Table 5.36. Results of SARAD for $NL1 = 1$, $NL2 = 1$ ($n = 50$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)	Avg.	Std. Dev.
50.25	1303.19	0	1	-	45.73	234	1291.84	18.82
50.26	1142.64	0	1	-	31.88	156	1119.61	39.36
50.27	1178.19	0	1	-	45.73	156	1173.29	6.40
50.28	1021.56	0	1	-	33.87	203	1004.50	31.55
50.29	1237.64	0	1	-	46.86	194	1213.34	24.76
50.30	1077.32	0	1	-	34.62	226	1076.78	1.13
50.31	1112.64	0	1	-	46.86	214	1062.67	85.72
50.32	952.41	0	1	-	32.56	269	952.39	0.04

As mentioned earlier, CPU times of SASCAD are significantly higher than SARAD. It is known that the CPU times increase exponentially as the problem size increases hence for 100-customer problem ($n = m = 100$) we only perform SARAD. The parameters for 100-customer problem are listed below:

Table 5.37. Parameters for 100-customer problems

max_*iter*	max_*initial*
80	4

Table 5.38. Costs for 100-customer problems

	f_1	f_2	ua_1	ua_2
L	30	120	4	12
H	70	280	6	20

Table 5.39. Results of SARAD for $NL1 = 2$, $NL2 = 2$ ($n = 100$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)
100.1	5656.10	1	1	65.69	95.78	1699
100.2	4790.14	0	3	-	24.30	964
100.3	5391.22	1	1	68.56	95.56	841
100.4	4626.04	1	1	43.43	66.76	1052
100.5	5514.25	0	1	-	123.27	2033
100.6	4803.08	1	2	30.82	35.26	1036
100.7	5364.62	1	1	64.39	99.90	1020
100.8	4668.72	1	1	50.63	66.66	1368

Table 5.40. Results of SARAD for $NL1 = 2$, $NL2 = 1$ ($n = 100$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)
100.9	6400.50	1	1	49.68	99.15	779
100.10	5579.79	0	1	-	79.52	1448
100.11	6163.48	0	1	-	110.45	992
100.12	5419.79	0	1	-	79.52	966
100.13	6449.69	3	1	27.48	71.43	1086
100.14	5859.82	1	1	50.95	56.35	1115
100.15	6318.40	0	1	-	100.23	807
100.16	5630.48	1	1	27.84	62.42	990

Table 5.41. Results of SARAD for $NL1 = 1$, $NL2 = 2$ ($n = 100$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)
100.17	6198.83	0	1	-	109.22	829
100.18	5472.70	0	1	-	76.65	1143
100.19	6038.83	0	1	-	109.22	1639
100.20	5312.70	0	1	-	76.65	1546
100.21	6281.39	0	1	-	96.63	906
100.22	5633.79	0	1	-	67.42	985
100.23	6121.39	0	1	-	96.63	896
100.24	5475.39	0	1	-	68.64	950

Table 5.42. Results of SARAD for $NL1 = 1$, $NL2 = 1$ ($n = 100$)

Prob. No.	TNP^*_{SARAD}	No.of Lev-1	No.of Lev-2	A_1	A_2	CPU (sec)
100.25	6506.90	0	1	-	109.47	1118
100.26	5771.32	0	1	-	78.46	2279
100.27	6346.90	0	1	-	109.47	941
100.28	5611.32	0	1	-	78.46	1110
100.29	6210.81	1	1	76.59	93.43	940
100.30	5509.26	1	1	64.79	65.83	1521
100.31	6223.42	1	1	83.05	90.55	849
100.32	5357.57	1	1	51.78	69.59	967

6. CONCLUSIONS and RECOMMENDATIONS FOR FUTURE RESEARCH

6.1. Summary and Conclusions

In this work, a competitive and hierarchical facility location problem has been studied. Because of the non-convexity of the objective function, standard solution techniques fail for these problems. Therefore some heuristics approach have been developed and tested on some test instances.

In the first strategy SASCAD, add-by-second-closest and drop-by-second-closest rules are applied for neighborhood search. The results are efficient however the strategy is time consuming. For small-sized problems CPU times are acceptable but as the size of the problem increases, CPU times increase dramatically. Random add-drop rules are employed in the second strategy namely SARAD. It was known that SARAD does not intend to improve the current solution in opposition to SASCAD. On the other hand it was expected that SARAD would be significantly faster than SASCAD.

In three different problem sizes, many instances are generated by defining different levels for number of competitor's facilities, the locations of competitors facilities, fixed costs of each level of facilities and unit attractiveness costs of each level of facilities. It is observed that in most of the experiments, SARAD returns results that are as good as the results of SASCAD. Moreover CPU times of SARAD are significantly less than CPU times of SASCAD. For small problem sizes both strategies are efficient and appropriate to employ whereas for large problem sizes, it is more sensible to use SARAD as the solution procedure. Two search algorithms are embedded in both algorithms to optimally find the attractiveness values. Revised Nelder Mead Simplex Search algorithm and Fibonacci Search algorithm are both convergence algorithms.

6.2. Recommendations for Future Research

Since it is known that location of competitive and hierarchical facilities is a rarely studied problem, researchers may easily extend the problem through many directions.

We assume that the number of each level of facilities to be opened are not predetermined. For simplification, a *pq* median version of this work can be studied. In this way utilizing only the swap moves might be efficient.

The continuous attractiveness variables are assumed to be a weighted sum of attributes, however we do not deal with the attributes one by one. As a future work it can be interesting to find the values of different attributes one by one. Furthermore the weights of the attributes may vary for each customer site.

From the hierarchical point of view, we develop a successively inclusive service hierarchy. Researchers may convert this assumption to a successively exclusive service hierarchy by taking private schools, some private hospitals, etc. into account.

To capture a more realistic approach, treating the costs as functions of locations may be beneficial.

Finally to see the effect of using a different heuristic algorithm, TS can be employed as a solution procedure for the same mathematical model. It is known that TS algorithm is a widely used method in facility location problems.

APPENDIX A: PSEUDOCODE OF SIMULATED ANNEALING ALGORITHM

Notation

$initial$	=	number of different initial solutions
$\max\_initial$	=	maximum number of different initial solutions
$no\_improvement$	=	number of non improvements
max_no_imp	=	maximum number of non improvements
$iter$	=	number of iterations preformed
max_iter	=	maximum number of iterations
obj_cur	=	objective value of current solution
obj_new	=	objective value of new solution
obj_best	=	best objective value among max_iter iterations
obj_best_imp	=	best objective value among max_no_imp iterations
loc_cur	=	facility locations of the current solution
$aval\_cur$	=	attractiveness values of the current solution
loc_new	=	facility locations of the new solution
$aval\_new$	=	attractiveness values of the new solution
loc_best	=	facility locations of obj_best
$aval\_best$	=	attractiveness values of obj_best
loc_best_imp	=	facility locations of the obj_best_imp
$aval\_best\_imp$	=	attractiveness values of the obj_best_imp
$intem$	=	initial temperature
$coef$	=	update coefficient of the temperature
$temp$	=	counter for temperature
$impro\_crt$	=	improvement criterion
$T(temp)$	=	temperature at the tempth iteration
$prev\_best\_imp$	=	the objective value of the previous improvement cycle
M	=	a sufficiently large number

Simulated Annealing algorithm

Input:

cd_{cj},bc_{cjk},d_{ij},BP_{jk},f_k,ua_k, CA_{ck}

Determine the parameters:

$coef$,max_no_imp,max_iter,$intem$,$impro\_crt$

$initial := 1$

`While`($initial < max\_initial$) `do`

Initialization

Generate an initial solution loc_cur, find *aval_cur*

calculate *obj_cur* by using SIMPLEX SEARCH

Set

$obj\_best\_imp := obj\_cur$

$loc\_best\_imp := loc\_cur$

$aval\_best\_imp := aval\_cur$

Set

$no\_improvement := 1$,$temp := 1$,$T(temp) := intem$

While($no\_improvement < max\_no\_imp$) do

Set $obj\_best := -M$

Set $iter := 1$

While($iter <$ max $\_iter$) do

Choose an operator randomly to determine *loc_new*

/* Among the operators listed below: */

1) Add level-1

2)Add level-2

3)Drop level-1

4)Drop level-2

5)Swap

If *loc_new* contains both level-1 and level-2 facilities then

GoTo SIMPLEX SEARCH to find *aval_new*

else

GoTo FIBONACCI SEARCH to find *aval_new*

Endif

Calculate obj_new

```
If obj_new > obj_cur then

obj_cur := obj_new

loc_cur := loc_new

aval_cur := aval_new

    If obj_new > obj_best then

    obj_best := obj_new

    loc_best := loc_new

    aval_best := aval_new

    Endif

else

SET

obj_cur := obj_new

loc_cur := loc_new

aval_cur := aval_new
```

WITH PROBABILITY $e^{(-(obj\_cur-obj\_new)/T(temp))}$

endif

$iter := iter + 1$

End While

If $obj\_best > obj\_best\_imp$ then

$obj\_best\_imp := obj\_best$

$loc\_best\_imp := loc\_best$

$aval\_best\_imp := aval\_best$

Endif

$T(temp + 1) := T(temp) * coef$

$temp := temp + 1$

If $(obj\_best\_imp - prev\_best\_imp)/(prev\_best\_imp + \varepsilon) < impro\_crt$ then

$no\_improvement := no\_improvement + 1$

else

$no\_improvement = 1$

endif

$prev\_best\_imp = obj\_best\_imp$

```
End while
```

$initial = initial + 1$

```
End while
```

REFERENCES

1. Aboolian, R., O., Berman, and D., Krass, "Optimizing Pricing and Location Decisions for Competitive Service Facilities Charging Uniform Price", *Journal of the Operational Research Society*, doi:10.1057/palgrave.jors.2602493, pp. 1-14, 2007.

2. Aboolian, R., O., Berman, and D., Krass, "Competitive Facility Location and Design Problem", *European Journal of Operational Research*, Vol.182, pp. 40-62, 2007.

3. Achabal, D. D., W. L., Gorr, and W. L., Mahajan, "MULTILOC: A multiple store location decision model", *Journal of Retailing*, Vol.2, pp. 5-25, 1982.

4. Bazaraa, M. S., H. D., Sherali, C. M., Shetty, "Nonlinear Programming Theory and Algorithms", *Wiley-Interscience,* 2005.

5. Benati, S. and P., Hansen, "The maximum capture problem with random utilities: Problem formulation and algorithms", *European Journal of Operational Research*, Vol.143, pp. 518-530, 2002.

6. Berman, O. and D., Krass, "Locating multiple competitive facilities: spatial interaction models with variable expenditures", *Annals of Operational Research*, Vol.111, pp. 197-225, 2002.

7. Brimberg, J., P., Hansen, and N., Mladenovic, E. D., Taillard, "Improvements and Comparison of heuristics for solving the uncapacitated Multisource Weber Problem", *Operations Research,* Vol.48, pp. 444-460, 2000.

8. Daskin, M. S., "Network and Discrete Location Models, Algorithms, and Applications", *New York: Willey*, 1995

9. Drezner, T., "Optimal Continuous Location of a Retail Facility, Facility Attractiveness, and Market Share", *Journal of Retailing*, Vol.1, pp. 49-64, 1994.

10. Drezner, T., "Location of multiple retail facilities with limited budget constraints-in continuous space", *Journal of Retailing and Consumer Services,* Vol.5 (3), pp. 173-184, 1998.

11. Drezner, Z.(Editor), "Facility Location: A Survey of Applications and Methods", *Springer*, 1995.

12. Drezner, T., and Z., Drezner, "Multiple Facilities Location in the Plane using the Gravity Model", *Geographical Analysis*, Vol.38, pp. 391-406, 2006.

13. Drezner, T., and Z., Drezner, "Lost demand in a competitive environment", *Journal of the Operational Research Society*, Vol.37, pp. 1-10, 2006.

14. Drezner, T., and Z., Drezner,"The gravity *p*-median model", *European Journal of Operational Research*, Vol.179, pp. 1239-1251

15. Drezner, T., and Z., Drezner, "Locating a single new facility among existing, unequally attractive facilities", *Journal of Regional Science*, Vol.2, pp. 237-252, 1994.

16. Drezner, T., and Z., Drezner, "Competitive facilities:Market Share and Location with Random Utility", *Journal of Regional Science*, Vol.36, pp.1-15

17. Drezner, T., and Z., Drezner, "Validating the gravity-based competitive location model using inferred attractiveness", *Annals of Operations Research*, Vol.111, pp. 227-237, 2002.

18. Drezner, T., and Z., Drezner, "Replacing Continuous Demand with Discrete Demand in a Competitive Location Model", *Naval Research Logistics*, Vol.44, pp. 81-95, 1997.

19. Drezner, T. and Z., Drezner, and S., Salhi, "Solving the multiple competitive facility location problem", *European Journal of Operational Research*, Vol.142, pp. 138-151, 2002.

20. Drezner, Z., and H. W., Hamacher,(Editors), "Facility Location: Applications and Theory", *Springer*, 2002.

21. Drezner, Z., G. O., Wesolowsky, T., Drezner, "On the logit Approach to Competitive Facility Location", *Journal of Regional Science*, Vol.38, Vol.2, pp. 313-327

22. Drezner, T., Z., Drezner, and S., Salhi, "A multi-objective heuristic approach for the casualty collection points location problem", *Journal of the Operational Research Society*, Vol.57, pp. 727-734, 2006.

23. Eiselt, H. A., and G., Laporte, "The Maximum Capture Problem In A Weighted Network", *Journal of Regional Science*, Vol.29 (3), pp. 433-439, 1989.

24. Fernandez, J., B., Pelegrín, F., Plastria, and B., Tóth, "Solving a Huff-like competitive location and design model for profit maximization in the plane", *European Journal of Operational Research*, Vol.179, pp. 1274-1287, 2007.

25. Fischer, K.,"Sequential Discrete *p*-facility Models for Competitive Location Planning", *Annals of Operations Research*, Vol.111, pp. 253-270, 2002.

26. Goodchild, M. F.,"ILACS:a Location Allocation Model for Retail Site Selection", *Journal of Retailing*, Vol.60, pp. 84-100, 1984.

27. Hakimi, S. L.,"On Locating New Facilities In a Competitive Environment", *European Journal of Operational Research*, Vol.12, pp. 29-35, 1983.

28. Hansen, P., D., Peeters, D., Richard, J. F., Thisse, "The Minisum and Minimax Location Problems Revisited", *Operations Research*, Vol.33 (6), pp. 1251-1265, 1985.

29. Hotelling, H.,"Stability in competition", *Economic Journal*, Vol.39, pp. 41-57, 1929.

30. Huff, D. L.,"Defining and estimating a trade area", *Journal of Marketing*, Vol.28, pp. 34-38, 1964.

31. Humphrey, D. G., J. R., Wilson, "A Revised Simplex Search Procedure for Stochastic Simulation Responce Surface Optimization", *Informs Journal on Computing*, Vol.12 (4), pp. 272-282, 2000.

32. Mahajan, V., A. K., Jain, B. T., Ratchford, "Use of Binary Attributes in the Multiplicative Competitive Interactive Choice Model", *Journal of Consumer Research*, Vol.5 (3), pp. 210-214, 1978.

33. McGarvey, R. G., T. C., Cavalier, "Constrained Location of competitive facilities in the Plane", *Computers and Operations Research*, Vol.32, pp. 359-378, 2005.

34. Nakanishi, M. and L. G., Cooper, "Parameter Estimate for Multiplicative Interactive Choice Model: Least Squares Approach", *Journal of Marketing Research*, Vol.11, pp. 303-311, 1974.

35. Narula, S. C., and U. I., Ogbu, "A hierarchical location-allocation problem", *Omega*, Vol.7, pp. 137-143, 1979.

36. Nelder, J. A., and R., Mead, "A simplex method for function minimization", *Computer Journal*, Vol.7, pp. 308-313, 1965.

37. Pirkul, H. and V., Jayaraman, "Production, transportation, and distribution planning in a multi-commodity tri-echelon system", *Transportation Science*, Vol.30, pp. 291-302, 1996.

38. Plastria, F. and E., Carrizosa, "Optimal location and design of a competitive facility", *Mathematical Programming*, Vol.2, pp. 247-265, 2004.

39. Plastria, F., "Static Competitive Facility Location: An Overview of optimisation approaches", *European Journal of Operational Research*, Vol.129, pp. 461-470, 2001

40. F.Plastria, "Profit maximising single competitive facility location in the plane", *Studies in Locational Analysis*, Vol.11, pp. 115-126, 1997.

41. Reilly, W. J., "The Law of Retail Gravitation", *Knickerbocker Press, New York*, 1929.

42. ReVelle, C., "The Maximum Capture or "Sphere of Influence" Location Problem:Hotelling revisited on a Network", *Journal of Regional Science*, Vol.26 (2), pp. 343-358, 1986.

43. Revelle, C. and D., Serra, "The Maximum Capture Problem Including Relocation", *INFOR (Canadian Journal of Operations Research)*, Vol.29, pp. 130-138, 1991.

44. Santos-Penate, D. R., R., Suarez-Vega, P., Dorta-Gonzales, "Competitive location in discrete space using attraction functions", *Paper presented at the EWGLA10 meeting, Murcia, Spain*, 1998.

45. Serra, D. and ReVelle, C. "Competitive Location in Discrete Space", in Drezner, Z.(editor),:"Facility Location: A Survey of Applications and Methods", *Springer*, 1995.

46. Serra, D. and C., ReVelle, "Market Capture by two Competitors:The preemptive Location Problem", *Journal of Regional Science*, Vol.34 (4), pp. 549-561, 1994

47. Serra, D. and C., ReVelle, "The *pq*-median problem:location and districting of hierarchical facilities", *Location Science*, Vol.1 (4), pp. 299-312, 1993.

48. Serra, D., V., Marianov, C., ReVelle, "The Maximum-capture hierarchical location problem", *European Journal of Operational Research*, Vol.62 (3), pp. 363-371, 1992.

49. Suzuki, T. and M. J., Hodgson, "Multi-Service Facility Location Models", *Annals of Operations Research*, Vol.123, pp. 223-240, 2003.

50. Suzuki, T. and M. J., Hodgson, "Optimal facility location with multi-purpose trip making", *IIE Transactions*, Vol.37, pp. 481-491, 2005.

51. Şahin, G. and H., Süral, "A Review of Hierarchical Facility Location Models",

Computers and Operations Research, Vol.34, pp. 2310-2331, 2007.

52. Teixeira, J. C., and A. P., Antunes, "A hierarchical location model for public facility planning", *European Journal of Operational Research*, Vol.185, pp. 92-104, 2008.

53. Weiszfeld, E., "Sur le point pour lequel la somme des distances de n points donnés est minimum", *Tohoku Mathematical Journal*, Vol.43, pp. 355-386, 1936.

54. Wendell, R. E. and R. D., McKelvey, "New perspectives in Competitive location theory", *European Journal of Operational Research*, Vol.6, pp. 174-182, 1981.