

A Web-Based Decision Support System for the Multiple Capacitated Facility Location Problem

Jason Papathanasiou¹, Nikolaos Ploskas^{2*}, and Nikolaos Samaras³

¹ University of Macedonia, 156 Egnatia Str., Thessaloniki 54006, Greece,
jasonp@uom.gr,

² University of Macedonia, 156 Egnatia Str., Thessaloniki 54006, Greece,
ploskas@uom.gr,

³ University of Macedonia, 156 Egnatia Str., Thessaloniki 54006, Greece,
samaras@uom.gr

Abstract. The facility location problem is a widely studied classical operations research problem. To address this problem we implement an algorithm that calculates the exact solution for a given multiple capacitated facility location problem so long as that exists. Many issues of this problem belong to the NP-hard class of algorithms and as a result the computation time is disappointing for large networks. Therefore, we present a dynamic approximation algorithm for the solution of this problem that is capable to compute an approximation solution in an acceptable time interval. The aforementioned algorithms are integrated in a web-based Decision Support System (DSS). The DSS offers the possibility to create either a random or a custom graph and to evaluate both algorithms by performing alternative scenarios for the future development of the market. Finally, the DSS can export the results of the evaluation to a Microsoft Word document for further use.

Key words: Decision Support Systems, Capacitated Facility Location Problem, Location Allocation Problem

1 Introduction

The facility location problem is a well-known operations research problem and has practical applications in many different fields [1] - [6]. In facility location-allocation models, enterprises are concerned with finding the best location to install their new facilities from a given set of potential sites; another set consists of existing and already established facilities with known locations [7] - [9]. The main aim of the new enterprises is to obtain the largest possible share and revenue from this specific market and at the same time to avoid any overlapping between the market segments that they will serve.

The problem has been studied extensively in the literature. It contains a wide range of proposed algorithms, variants and extensions [10] - [12]. Many approximation algorithms have been also proposed [13] - [18]. In this paper, we consider

* Corresponding author

a particular type of the facility location problem, called the Multiple Capacitated Facility Location Problem (MCFLP) under time constraints. To outline the problem, a number of existing competitive enterprises, which produce certain products or services, operate in a specific market. The market requires a specific quantity of this product or a specific level of this service in a determined time period and the already existing enterprises cover the needs of the market to the greatest degree. A number of new cooperating enterprises attempt to enter the market and seek the best locations from a number of available candidate locations. The new enterprises seek to acquire the largest possible share of this market and avoid any overlapping between the market segments served. Hence, the locations of these entering enterprises must ensure their economic viability. Each new enterprise ideally should occupy an adequate share of the market, so that its production is higher than a determined sales threshold level [19]. In other words, the enterprises aim to ensure their economic viability in order to make such an investment. Existing enterprises also aim to be economically viable; if not, they will be taken off the map [20][21]. The market surface can be simulated by a network with existing facilities nodes, demand nodes and candidate nodes.

This paper is an extension of the work of Papathanasiou and Manos [22]. We present two algorithms for the solution of a given MCFLP: (i) an algorithm that finds the exact solution of the problem so long as this exists, and (ii) a dynamic approximation algorithm that can calculate an approximation solution in an acceptable time interval. These algorithms have been proposed by Papathanasiou and Manos [22]. The innovation of this paper is that we integrate these algorithms in a web-based DSS that can assist policy makers find the best locations for their enterprises.

Only few software packages exist for the solution of this problem exclusively. In the present paper, a web-based DSS for the MCFLP is presented. The proposed algorithms have been implemented using MATLAB and converted to Java classes, while the web interface of the DSS has been implemented using Java. The DSS offers the possibility to create either a random or a custom graph. The custom graph is inserted from a valid Microsoft Excel or text file. After the creation of the market's network, users can evaluate both algorithms by performing alternative scenarios for the future development of the market. Finally, the DSS can export the results of the evaluation, including tables and animated diagrams to a Microsoft Word document for further use.

The structure of the paper is as follows. Section 2 presents the background of our work. Section 3 includes the presentation of the two algorithms. In Section 4, a web-based DSS for the MCFLP is presented. Finally, the conclusions of this paper are outlined in Section 5.

2 Model Specification

The mathematical form of the problem described in Section 1 can be formulated as follows (for a more detailed description, see [22]):

$$\max \sum_i \sum_p DP_{ip} X_i \tag{1}$$

or

$$\max \sum_i \sum_p a DP_{ip} Q_{ip} X_i \tag{2}$$

subject to

$$DP_{ipmin} \leq DP_{ip} \leq DP_{ipmax} \tag{3}$$

$$\sum_i X_i = P \tag{4}$$

$$Y_{ij} - X_i \leq 0 \tag{5}$$

$$X_i = 0, 1 \tag{6}$$

$$Y_{ij} = 0, 1 \tag{7}$$

$$UP_{ij} = 0, 1 \tag{8}$$

$$UM_{mj} = 0, 1 \tag{9}$$

$$\sum_p DP_{ip} = \sum_p \sum_i \sum_j H_{ij}^p Y_{ij} UP_{ij} \tag{10}$$

where:

$|P|$: the cardinality number of new enterprises

$$p_n \in P = \{p_1, p_2, \dots, p_k\}, n = 1, 2, \dots, k$$

$|M|$: the cardinality number of existing enterprises

$$m_f \in M = \{m_1, m_2, \dots, m_h\}, f = 1, 2, \dots, h$$

$|I|$: the cardinality number of candidate nodes of new enterprises

$$i_s \in I = \{i_1, i_2, \dots, i_q\}, i = 1, 2, \dots, q$$

$|J|$: the cardinality number of demand nodes

$$j_r \in J = \{j_1, j_2, \dots, j_b\}, j = 1, 2, \dots, b$$

T : the time within which the market demands a specific quantity of the product in question

DP_{ip} : the production capacity in time T of the new enterprise p established in node i

DP_{ipmax} : the maximum production capacity in time T of the new enterprise p established in node i

DP_{ipmin} : the minimum acceptable production capacity in time T of the new enterprise p established in node i

DM_m : the production capacity in time T of the existing enterprise m

DM_{max} : the maximum production capacity in time T of the existing enterprise m

DM_{min} : the minimum acceptable production capacity in time T of the existing enterprise m

H_j : demand in demand node j

HP_{ij}^p : the fraction of demand in node j , which is serviced by node i where the new enterprise p has been located

HM_{mj} : the fraction of demand in node j where the existing enterprise m has been located

S_{pi} : the range of new enterprise p in node i and in time T

S_m : the range of existing enterprise m in time T

Q_{ip} : the production cost of new enterprise p in node i .

Q_m : the production cost of existing enterprise m .

a : the profit percentage.

$$X_i = \begin{cases} 1, & \text{if the new enterprise is located at the candidate node } i, \\ 0, & \text{if not} \end{cases}$$

$$Y_{ij} = \begin{cases} 1, & \text{if the demand in node } j \text{ is serviced by a new enterprise in node } i, \\ 0, & \text{if not} \end{cases}$$

$$UP_{ij} = \begin{cases} 1, & \text{if node } j \text{ is within the range of } i \text{ in time } T, \\ 0, & \text{if not} \end{cases}$$

$$UM_{mj} = \begin{cases} 1, & \text{if node } j \text{ is within the range of } m \text{ in time } T, \\ 0, & \text{if not} \end{cases}$$

The total number of nodes of the network is $|I|+|J|+|M|$. Objective function (1) refers to the maximization of the product that was produced, in the event that the cooperating enterprises choose the aggressive tactic, while objective function (2) deals with the conservative approach of the maximization of profit.

3 Algorithms' Presentation

In this section, the two algorithms for the solution of a given MCFLP are presented: (i) an algorithm that finds the exact solution, and (ii) a dynamic approximation algorithm that can calculate an approximation solution. These algorithms have been proposed by Papathanasiou and Manos [22].

Table 1. Exact Algorithm

<p>Step 1. Locate the new enterprises p_n at random positions from the set of candidate nodes i_s.</p> <p>Step 2. Share the demand of each demand node j_r to the new and existing enterprises, p_n and m_f, respectively, with the lowest sales value GP_{pij} and GP_{mj} as a criterion.</p> <p>Step 3. Find the enterprises p_n and m_f that are economically viable (an enterprise is economically viable if $DP_{ip} \geq DP_{ipmin}$ and $DM_m \geq DM_{mmin}$, respectively. If all enterprises are economically viable, then go to Step 7.</p> <p>Step 4. If one or more new enterprises p_n are not economically viable, then the enterprise which has the greatest difference $DP_{ipmin} - DP_{ip}$ moves to another position from the set of the candidate nodes i_s. If it is not possible, then the enterprise which has the next greatest difference $DP_{ipmin} - DP_{ip}$ moves. Demand is re-shared and the algorithm goes to Step 2. If all possible combinations of new enterprises p_n in candidate nodes i_s have been tested, then the problem is rendered impossible and the algorithm stops.</p> <p>Step 5. If one existing enterprise m_f is not economically viable, its demand is re-shared and is taken off the map. If there is still another existing enterprise that is not economically viable, then the Step 5 is repeated until all remaining existing enterprises are economically viable.</p> <p>Step 6. If one or more new enterprises p_n are not economically viable, then algorithm goes to Step 4.</p> <p>Step 7. Calculate the objective value and if it is better than the previous one, keep this value.</p> <p>Step 8. The new enterprise p_n which has the greatest difference $DP_{ipmin} - DP_{ip}$ moves to another position. If it is not possible, then the enterprise which has the next greatest difference $DP_{ipmin} - DP_{ip}$ should be moved and the algorithm goes back to Step 2.</p> <p>Step 9. Repeat Steps 2-8, until no further improvement of the objective function is possible or all possible combinations of new enterprises p_n in candidate nodes i_s have been tested.</p>
--

4 Decision Support System Presentation

Figure 1 presents the decision making process that the decision-policy maker can perform using the DSS. Firstly, the decision maker formulates the problem under examination. Then, the data acquisition, validation and verification phase follows, so, the decision maker may upload the input data to the DSS and preview the market mapping and analysis, in the next step. Then, the algorithms' evaluation and execution step follows. The last step includes the presentation

Table 2. Dynamic Approximation Algorithm

<p>Step 1. Find the economically and time distances for each demand node j_r from each candidate node i_s.</p> <p>Step 2. Each demand node j_r consumes from its nearest in terms of time node and only from that. If a demand node j_r is equidistant to two or more nodes, then demand node j_r consumes from its nearest economical node.</p> <p>Step 3. Candidate nodes i_s are classified based on the number of demand nodes j which are assigned to them.</p> <p>Step 4. For the candidate nodes i_s, the total demand which they serve is found. The new enterprise p_n which has the greatest possible production at node i_s with the highest total demand is located at that node. Then, the new enterprise p_n which has the next greatest possible production is located at one of the remaining nodes i_s with the highest total demand is located at that node and the process is repeated until all enterprises p_n have been assigned to a candidate node i_s.</p> <p>Step 5. Find if the solution is acceptable; if it is, the algorithm terminates.</p> <p>Step 6. If not, the new enterprise p_n which is economically viable and has the biggest difference between the total demand of demand nodes j_r that it serves and its minimum production at that candidate node i_s. Furthermore, the demand node j_r, which is the closest to the enterprises p_n that do not survive and which has the biggest respective difference, is withdrawn from the total of demand nodes which that unit p_n serves. If no unit p_n survives, then then the location cannot be found and the algorithm stops.</p> <p>Step 7. Find if the solution is now acceptable. If it is, then the algorithm terminates, otherwise go back to Step 6.</p>
--

and the analysis of the results. Finally, the decision maker validates the results and if necessary provides feedback on the operation and the updated decision making process is performed again.

**Fig. 1.** Decision Making Process

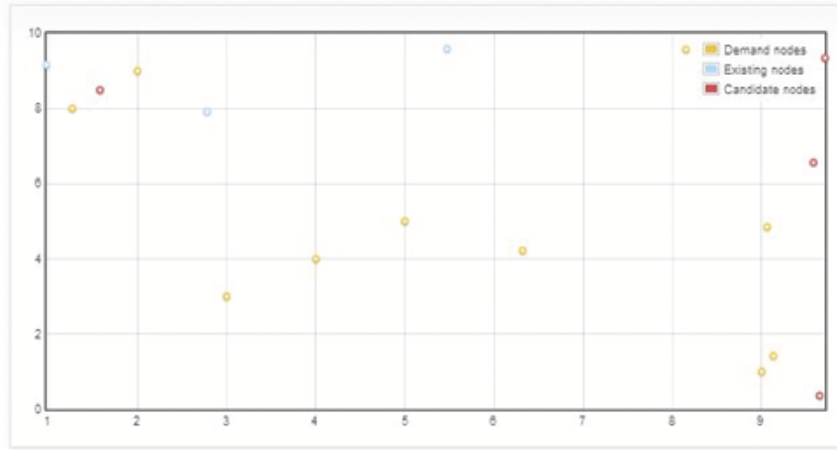
Both algorithms have been implemented using MATLAB R2013a and have been converted to Java classes using the MATLAB Builder JA version 2.2.6. The web interface of the DSS has been designed using jsp and multiple users can access it through a web browser.

The screen for creating random graphs is shown in Figure 2. The decision maker can fill out the form or press the 'Load Default Values' in order to create a random graph. Furthermore, the decision maker can press the 'Upload Data From Excel File' option in order to upload an excel file with the appropriate data (Data input step in the decision making process). An excel template file is available to the policy makers in order to format their data. Then, a screen with the market representation is presented, as shown in Figure 3 (Market mapping and analysis step in the decision making process). By pressing the 'Execute algorithms and collect the results' option (Algorithms evaluation and execution step in the decision making process), the report page with the results is presented, as shown in Figures 4 - 5 (Results presentation and analysis step in the decision making process). For each algorithm the following information is available: (i) total product consumed by the demand nodes, (ii) the execution time needed to solve the given problem, (iii) the number of iterations performed by the algorithm, and (iv) a figure showing the allocation of the demand nodes to the existing and new enterprises. Figures 3 - 5 present a case study with 10 demand nodes, 3 existing enterprises, 4 candidate nodes and 2 new enterprises. Both algorithms found the same solution where all new enterprises and one existing enterprise are economically viable; the other two existing enterprises do not survive and will be taken off the map. The dynamic approximation algorithm is 5 times faster and performs 4 times less iterations than the exact algorithm. In Figures 4 - 5, the lines connecting the red points with the blue points represent the demand nodes that are served by existing enterprises. The lines connecting the yellow points with the blue points represent the demand nodes that are served by new enterprises.

Step 1: Create a Random Graph or [Upload Data From Excel File](#)

Number of New Enterprises:	1
Number of Candidate Nodes:	1
Number of Existing Enterprises:	1
Number of Demand Nodes:	5
Area length:	
Max radius that enterprises can reach:	
Time spent for every 10 kilometers:	
Node capacities (lower and upper bounds):	Lower bound
	Upper bound
Production cost per unit in every candidate location (lower and upper bounds):	Lower bound
	Upper bound
Profit per unit in every candidate location (lower and upper bounds):	Lower bound
	Upper bound
Profit per unit for each existing enterprise (lower and upper bounds):	Lower bound
	Upper bound
Product demand for each demand node (lower and upper bounds):	Lower bound
	Upper bound
Minimum production capacity in every candidate location (lower and upper bounds):	Lower bound
	Upper bound
Load Default Values	
Create Random Graph	

Fig. 2. Screen for creating random graphs



[Execute algorithms and collect the results](#)

Fig. 3. Market representation

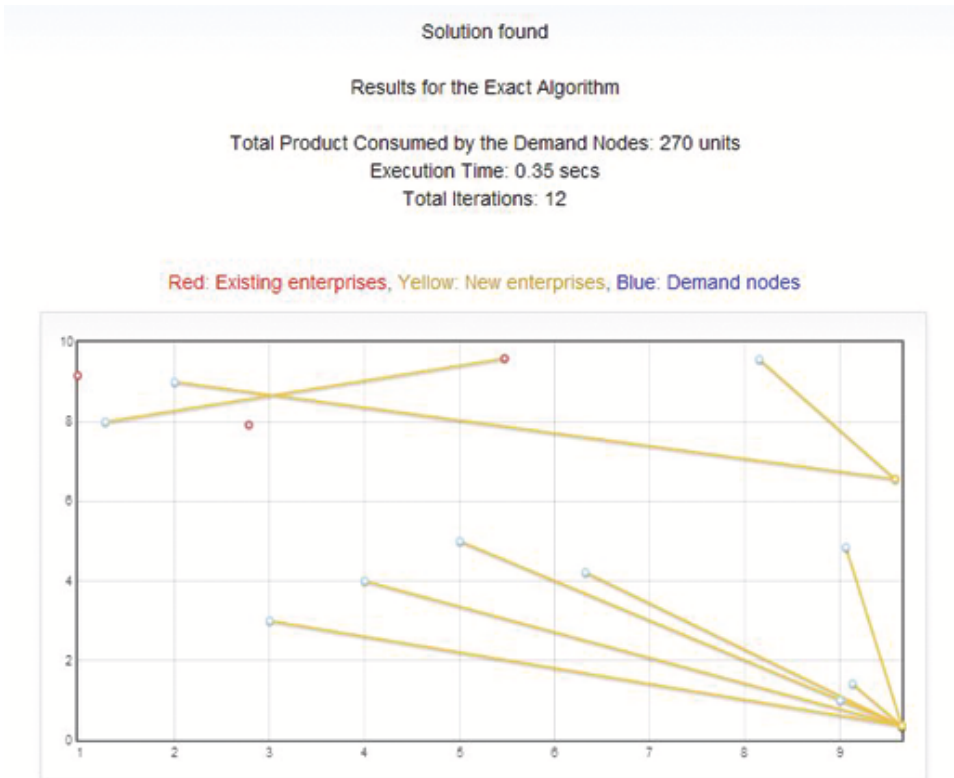


Fig. 4. Report of the Results for the algorithm that finds the exact solution

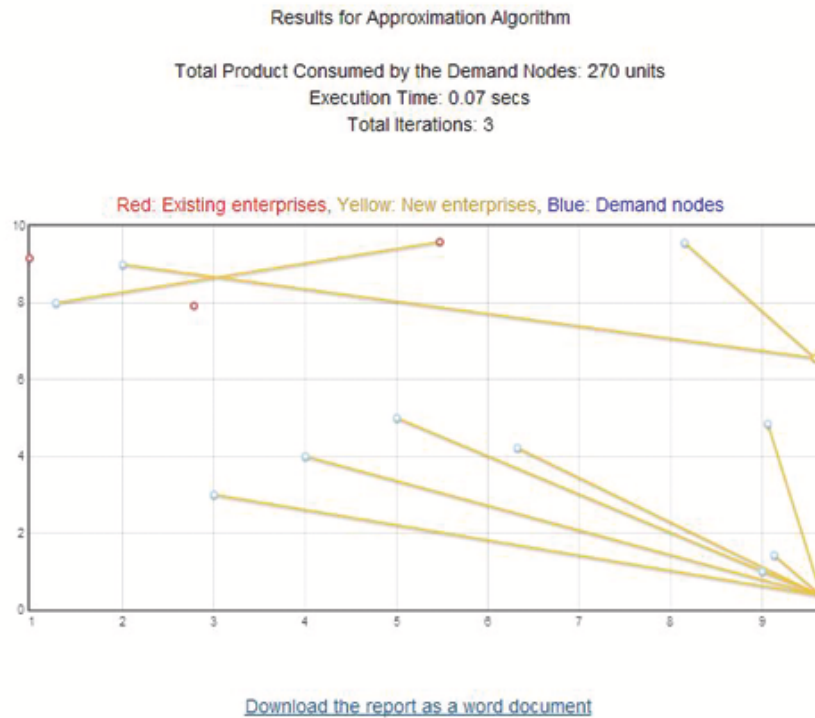


Fig. 5. Report of the Results for the dynamic approximation algorithm

A computational study was also conducted in order to highlight the speedup obtained from the dynamic approximation algorithm over the algorithm that calculates the exact solution. Table 3 presents the execution times of the two algorithms for 4 new enterprises, 7 existing enterprises, 7 candidate nodes and varying numbers of demand nodes.

The proposed web-based DSS has important managerial implications. First, decision makers can formulate their case studies and get a thorough analysis on if their enterprises should enter a market or not. The upload of the input data is a straight-forward procedure and the report of the results is user-friendly and and comprehensive. Moreover, the decision makers obtain an overview of the accuracy and performance of the dynamic approximation algorithm.

Some limitations exist on the proposed DSS. First, some input data referring to the existing enterprises and the demand nodes may not be available to the decision makers. A second potential limitation of the proposed DSS is that it does not provide information about the improvements that the new enterprises should do in order to either obtain the largest possible share and revenue from a specific market or be economically viable, if they are already not.

Table 3. Execution times and speedup obtained from the dynamic approximation algorithm over the algorithm that calculates the exact solution for 4 new enterprises, 7 existing enterprises, 7 candidate nodes and varying numbers of demand nodes

Demand Nodes	Algorithm that Calculates the Exaxt Solution (sec)	Dynamic Approximation Algorithm (sec)	Speedup
10	0.81	0.22	3.68
20	2.10	0.45	4.67
30	4.22	0.88	4.80
40	7.48	1.52	4.92
50	12.34	2.57	4.80
60	20.03	4.12	4.86
70	28.55	6.04	4.73
80	37.89	8.20	4.62
90	46.80	10.48	4.47
100	57.12	12.75	4.48
Average	21.73	4.72	4.60

5 Conclusions

In this paper, we presented two algorithms for the solution of the MCFLP. The first algorithm finds the exact solution of the given problem, while the second is a dynamic approximation algorithm that calculates an approximation solution in a small time interval. Although, the facility location problem is a classic operations research problem and many algorithms have been proposed for its solution, only few software packages exist exclusively for the solution of this problem. This paper presented a web-based DSS that assists decision-policy makers in the evaluation of their enterprises' economic viability when entering a new market. All algorithms have been implemented using MATLAB, while the web interface of the DSS has been designed using jsp and users can access it through a web browser from their PC/laptop or their smart device.

In future work, we plan to enhance the DSS with other options that will give decision makers the opportunity to get some alternative scenarios to investigate in order to obtain the largest possible share and revenue from a specific market. Finally, we plan to present real application case studies for which the proposed DSS can be utilized.

References

1. Brandeau, M.L., Chiu, S.S.: An overview of representative problems in location research. *Management Science* 35:6, 645-674 (1989)
2. Daskin, M.S.: *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley, New York (1995)
3. Drezner, Z., Hamacher, H.W.: *Facility Location: Theory and Algorithms*. Springer, Berlin (2001)

4. Marianov, V., Serra, D.: Location problems in the public sector. In: Drezner, Z., Hamacher, H.W. (eds.) *Facility Location: Applications and Theory*, Springer, Berlin (2002)
5. Melo, M., Nickel, S., Saldanha da Gama, F.: Facility location and supply chain management—a review. *European Journal of Operational Research* 196:2, 401–412 (2009)
6. Reville, C., Eiselt, H., Daskin, M.S.: A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research* 184:3, 817–848 (2008)
7. Aboolian, R., Berman, O., Krass, D.: Competitive facility location and design problem. *European Journal of Operational Research* 182:1, 40–62 (2007)
8. Drezner, T., Drezner, Z., Salhi, S.: Solving the multiple competitive facilities location problem. *European Journal of Operational Research* 142, 138–151 (2002)
9. Plastria, F.: Static competitive facility location: An overview of optimization approaches. *European Journal of Operational Research* 129, 461–470 (2001)
10. Aardal, K.: Capacitated facility location: separation algorithm and computational experience. *Mathematical Programming* 81, 149–175 (1998)
11. Owen, S.H., Daskin, M.S.: Strategic facility location: a review. *European Journal of Operational Research* 111, 423–447, (1998)
12. Sridharan, R.: Invited review the capacitated plant location problem. *European Journal of Operational Research* 87, 203–213 (1995)
13. Avella, P., Boccia, M., Sforza, A., Vasilev, I.: An effective heuristic for large-scale capacitated facility location problems. *J. Heuristics* 15, 597–615 (2009)
14. Beasley, J.E.: Lagrangean heuristics for location problems. *Eur. J. Operat. Res.* 65, 383–399 (1993)
15. Chudak, F.A., Williamson, D.P.: Improved approximation algorithms for capacitated facility location problems. In: 7th International IPCO Conference Proceedings, 99–113 (1999)
16. Cornuejols, G., Sridharan, R., Thizy, J.M.: A comparison of heuristics and relaxations for the capacitated plant location problem. *Eur. J. Oper. Res.* 50, 280–297 (1991)
17. Korupolu, M., Plaxton, C., Rajaraman, R.: Analysis of a local search heuristic for facility location problems. *J. Algorithms* 37, 146–188 (2000)
18. Kuehn, A.A., Hamburger, M.J.: A heuristic program for locating warehouses. *Manag. Sci.* 9, 643–666 (1963)
19. Shonwiler, J., Harris, T.: Rural Retail Business Thresholds and Interdependencies. *Journal of Regional Science* 21, 189–198 (1996).
20. Serra, D., Reville, C., Rosling, K.: Surviving in a competitive spatial market: The threshold Capture Model. *Journal of Regional Science* 4:39, 637–652 (1999)
21. Shiode, S., Drezner, Z.: A competitive facility location problem on a tree network with stochastic weights. *European Journal of Operational Research* 149, 47–52 (2003)
22. Papathanasiou, J., Manos, B.: An approximation algorithm for the location of dairy enterprises under time constraints. *European Journal of Operational Research* 182:3, 1479–1487 (2007)