Location problems solving by spreadsheets

WEN-CHEN LEE*, NENG-SHU YANG
Graduate School of Industry Engineering and Management
National Yunlin University of Science and Technology
123 University Road, Section 3, Douliou, Yunlin 64002
TAIWAN
g9421806@yuntech.edu.tw, yangns@yuntech.edu.tw

Abstract: Location problems have been the focus of considerable attention both in research and practice for many years. Numerous extensions have been described to broaden its appeal and enhance its applicability. This study proposes a spreadsheet approach for three well-known facility location problems, the *p*-median problem (PMP), capacitated *p*-median (CPMP), and maximal covering location problem (MCLP). The advantages of the method are not only relatively simple but are also effectively used, maintained and updated by users. The results show that the spreadsheet approach can generate good solutions, including conditions descriptions of location and allocation, within a reasonable process time. For some instances, the solutions obtained by this method are better than those by state of the art approaches. In addition, most location problems can also apply this approach.

Key-Words: location problems, *p*-median problem, capacitated *p*-median problem, maximal covering location problem, spreadsheets

1 Introduction

The logistics for distribution of products (or services) has been a subject of increasing importance over the years. It is a significant part of the strategic planning of both public and private enterprises. Decisions concerning the best configuration for the installation of facilities in order to attend demand requests are the subject of a wide class of problems, known as *location problems* [1]. These location problems have received a considerable amount of attention from scientists who have identified various problem types and developed variety methodologies to solve these problems, subsequently being adopted to make decisions belonging to locations of facilities in many practical applications.

Brandeau and Chiu [2] presented a review of representative problems that have been studied in location research and identify more than fifty problem types and indicate how those problem types relate to one another. The location problems can be described as models in which a number of facilities is to be located in the presence of customers, so as to meet some specified objectives. Obvious applications of the problem occur when facilities such as warehouses, plants, hospitals, or fire stations are to be located. Although these instances are quite different from each other, they share some common features.

Most location problems can be defined as follows: given space, distance, a number of customers, customers' demands and mission. The distance is defined between any two points in that area. The number of customers is located in the area under consideration and who have a certain demand for a product (or service). The mission is to locate one or more facilities in that area that will satisfy some or all of the customers' demands.

Depending on the objectives, location problems can be grouped into two major classes [3]. One class treats the minimization of the average or total distance between customers and facilities. The classic model that represents the problems of this class is the *p*-median problem (PMP). Optimally locating public and private facilities such as schools, parks and distribution centers are typical examples of this problem. The other class deals with the maximum distance between any customer and the facility designed to attend the associated demand. They often used in applications related the location of emergency facilities. These problems are known as covering problems and the maximum service distance is covering distance.

The *p*-median is a well-known facility location problem which addresses the supply of a single commodity from a set of potential facility sites to a set of customers with known demands for the commodity. The problem consists of finding the locations of the facilities and the flows of the commodity from facilities to customers such that transportation costs are minimized. The PMP was first introduced by Hakimi [4]. It is NP-hard, and so optimal solutions to large sized problems are difficult to obtain.

The PMP has received a considerable amount of attention from researchers who have developed a variety of solution procedures to solve it. Several exact and heuristic solution procedures have been developed for PMP. The exact procedures include the algorithms of Narula et al. [5]; Galvão [6] and Galvão and Raggi [7] among others. Heuristic solution procedures started with the paper of Teitz and Bart [8]. Pizzolato [9] developed a heuristic for large weighted graphs and applied it to locate schools in the metropolitan area of Rio de Janeiro, Brazil. A variety of metaheuristic approaches for the PMP have been proposed in recent years. These include the two stage construction heuristic [10]; the tabu search procedures [11]; the variable neighborhood search approaches [12]; the statistical analysis of simulated annealing [13] and the genetic algorithm [14,15].

The capacitated p-median problem (CPMP) considers capacities for the service to be given by each median, and the total service demanded by vertices identified by *p*-median clusters cannot exceed their service capacity. This problem is also known to be NP-hard and consists of finding the set of *p* medians and the assignment pattern that satisfies the capacity constraints with a minimum total cost. As a matter of fact, the CPMP is not intensively studied as the classical PMP. Recent approaches apply metaheuristics, such as simulated annealing and tabu search [16]; genetic algorithms [17]; scatter search and path relinking [18]. Pirkul et al. [19] described a visual interactive decision support tool, VisOpt, for CPMP. In a 2006 paper Reese [20] summarized the literature on solution methods for the uncapacitated and capacitated *p*-median problem on a network.

The maximal covering location problem (MCLP) proposed by Church and ReVelle [21] is a well-studied problem in the category of models that provide coverage to demand areas. MCLP does not require that all demand areas be covered; rather the objective is to locate *p* facilities such that the maximal population is covered within the service distance. Church and ReVelle proposed the linear programming relaxation of the 0-1 integer programming formulation of the problem and greedy-interchange heuristics to solve the MCLP. Schilling et al. [22] provide a detailed review of the covering models in facility location. Galvão and ReVelle [23] developed a lagrangean heuristic for the MCLP. Dwyer and Evans [24] developed an exact method for the particular case where all demand areas have equal weights. Downs and Camm [25] have reported an extensive computational evaluation of their method, dual-based algorithm, in terms of both variety of applications and problem size. More recently, Lorena and Pereira [26] present results obtained with a lagrangean/surrogate heuristic using a subgradient optimization method. Maric et al. [27] used genetic algorithm to find out the solutions of the two-level hierarchical covering location Problem.

It is hard to select an optimization tool that could not only be more easily understood but also effectively used, maintained and updated by users. Spreadsheets are inherently form-free and impose no particular structure on the way problems are modeled. All of the major spreadsheet products (e.g. Excel, Lotus, and Quattro) are supplied with optimization capability. Most commercial spreadsheets contain a “Solver” tool that requires no code to be written, and requires little knowledge about the optimization algorithms themselves, and so are extremely easy to use. The main advantage of these software packages is their wide availability and ease of use because they do not require custom coding. Traditional dedicated OR software packages (e.g. Lindo, GAMS, AMPL, CPLEX) have quite the opposite characteristic and impose fairly rigid rules or structures for modeling problems, not to mention the need to learn an algebraic modeling language. Spreadsheet models allow us to build more detailed and more complex models than traditional mathematics allows. In spite of their huge popularity, little has been written about how one should develop an optimization model using spreadsheets. In addition, they also have the advantage of being pervasive in problem analysis. Therefore, this present study built spreadsheet models rather than using a specialized mathematical modeling package.

Since their introduction over 15 years ago, electronic spreadsheet programs like Excel have become the most common tool business people use to model and analyze quantitative problems [28]. Many authors consider spreadsheets the tool of choice for today’s managers since they provide a convenient way for business people to build computer models, including optimization models [29]. Rasmussen [30] indicated that spreadsheet solvers could very well be the preferred software for solving quadratic assignment problem, compared to other general purpose optimization software. Based on the above reasoning, this work set out to develop a spreadsheet method to solve the location problems.

2 The Problem Formulations

The mathematical models of the PMP, CPMP and MCLP are presented as following. They have all been reformulated by this work to fit spreadsheet models.

2.1 The PMP model

The *p*-median model is a location/allocation model, which locates *p* facilities among *n* demand points and allocates the demand points to the facilities. The objective is to minimize the total demand-weighted distance between the demand points and the facilities. For an *n* vertex network and a symmetric distance matrix D = $[d_{ij}]_{n\times n}$, the PMP can be formulated as the following binary integer programming problem:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} w_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad \forall i \quad (2)$$

$$x_{ij} \le y_j \qquad \forall i, j \quad (3)$$

$$\sum_{i=1}^{n} y_j = p \quad (4)$$

$$x_{ij} = \{0, \ 1\} \qquad \forall i, j \quad (5)$$

$$y_j = \{0, \ 1\} \qquad \forall j \quad (6)$$

where

n = total number of demand points

w_i = demand at point i

d_{ij} = travel distance between point i and j

x_{ij} = 1 if point i is assigned to facility located at point j; 0 otherwise

y_j = 1 if facility is located at node j; 0 otherwise

p = number of facilities to be located

This is an uncapacitated facility location model where every demand point is served by one facility and trips to demand points are not combined. The constraints in a standard *p*-median model are fairly simple. Each demand point must be assigned to exactly one median (facility), represented by (2). Constraints (4) guarantee that exactly *p* medians are located. The computational difficulties regarding optimality are made complex by the fact that the variables are binary variables. The PMP is an integer programming problem, and constraint (5) and (6) provide the integer conditions.

2.2 The CPMP model

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} w_i d_{ij} x_{ij} \quad (1)$$

s.t. (2), (3), (4), (5), and (6)

$$\sum_{i=1}^{n} w_i x_{ij} \le c_j y_j \qquad \forall j \quad (7)$$

where

c_j = capacity of facility (or point) j

This is a CPMP model where every demand point is served by one facility and total median capacity must be respected. The objective of CPMP is also to minimize the total demand-weighted distance between the demand points and the facilities as well. Constraint (7) ensures that the capacity of every selected median is not exceeded, and the other constraints ((2), (4), (5), and (6)) are the same with the standard *p*-median model.

2.3 The MCLP model

The MCLP model is also a location/allocation model. The mathematical programming of the MCLP can be formulated as below:

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} w_i s_{ij} x_{ij} \quad (8)$$

s.t. (2), (3), (4), (5), and (6)

where

n = total number of demand points

w_i = demand at point i

d_{ij} = travel distance between point i and j

x_{ij} = 1 point i is covered by facility located at point j; 0 otherwise

y_j = 1 if facility is located at node j; 0 otherwise

p = number of facilities to be located

d_s = distance beyond which a demand area is considered uncovered

s_{ij} = 1 if $d_{ij} \le d_s$; 0 otherwise

The objective function (8) maximizes the coverage level within the maximum critical distance, d_s. The constraint (2), (3), (4), (5), and (6) impose the equal restriction as the *p*-median model. The MCLP is also an integer programming and NP-hard problem which means very difficult to solve as the number of *n* is large.

3 A simple spreadsheet example

A problem from [31] with n = 5 and p = 3 was used to provide a numerical instance. This work proposes three small examples to show how to solve and view the results for these problems.

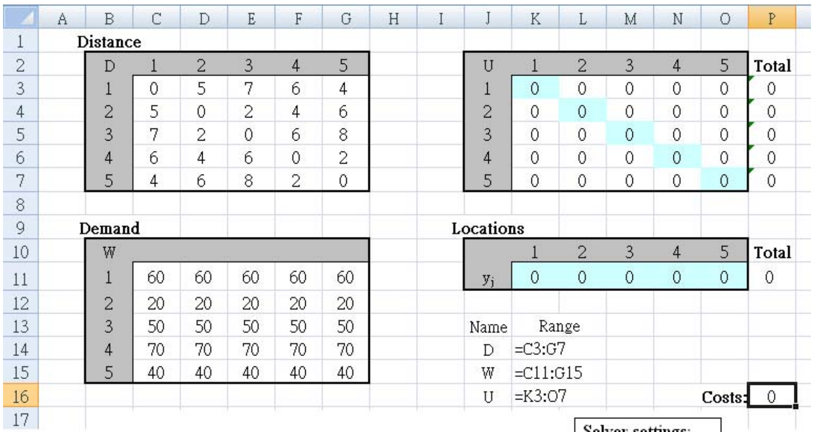
3.1 The PMP example

In the general case, it has a symmetric matrix $D_{[n\times n]}$ of distances between nodes i and j, and a matrix $W_{[n\times n]}$ expanded from $W_{[n\times 1]}$ of demands of node i with a total of n nodes. Let $i, j \in \{1, . . . , n\}$ be indexes for the nodes. Finally, defining the matrix of decision variables $U_{[n\times n]}$ as x_{jj}=1 if a facility is located in node j; otherwise x_{jj}= 0. In addition, x_{ij}=1 ($i \neq j$) if a node i is assigned to facility j; otherwise x_{ij}= 0. The computation of the costs as described in Eq. (1) and there is a simpler way of representing the total cost C:

$$\min C = \min \sum_{i=1}^{n} \sum_{j=1}^{n} w_i d_{ij} x_{ij} = \text{SUMPRODUCT}(W\,.\;D\,.\;U) \quad (9)$$

Figure 1 shows the binary representation of the PMP. The cost function from Eq. (9) is implemented, and it therefore can use the *Standard LP/Quadratic Solver* engine provided by Excel. The Solver settings are also shown in Figure 1. The matrix U was started with a null matrix (all entries zero). A SUMPRODUCT function is then used to compute the matrix W, D and U. SUMPRODUCT is one of the most versatile functions provided in Excel. In its most basic form, SUMPRODUCT multiplies corresponding members in given arrays, and returns the sum of those products. The demand is transformed into the matrix W because SUMPRODUCT only deal with the same size of arrays.

The result is shown in Figure 2. The minimum total demand-weighted distance is 120 which is the same as the result given by [31]. In addition, three facilities are located at node 1, 3 and 4; node 2 and 5 are assigned to facility located in 3 and 4 respectively.



Cell	Formulas	Copied to
P16	=SUMPRODUCT(D,W,U)	
P3	=SUM(K3:O3)	P4:P7
K11	=K3	
L11	=L4	
M11	=M5	
N11	=N6	
O11	=O7	
P11	= SUM(K11:O11)	

Solver settings:
Min: P16
By changing: U
Subject to:
U = Binary
P3:P7 = 1
K3:K7 <= K3
L3:L7 <= L4
M3:M7 <= M5
N3:N7 <= N6
O3:N7 <= O7
P11 = p

Figure 1. A small example of the PMP

Distance

D	1	2	3	4	5
1	0	5	7	6	4
2	5	0	2	4	6
3	7	2	0	6	8
4	6	4	6	0	2
5	4	6	8	2	0

U	1	2	3	4	5	Total
1	1	0	0	0	0	1
2	0	0	1	0	0	1
3	0	0	1	0	0	1
4	0	0	0	1	0	1
5	0	0	0	1	0	1

Demand

W					
1	60	60	60	60	60
2	20	20	20	20	20
3	50	50	50	50	50
4	70	70	70	70	70
5	40	40	40	40	40

Locations

	1	2	3	4	5	Total
y_j	1	0	1	1	0	3

Name	Range
D	=C3:G7
W	=C11:G15
U	=K3:O7

Costs: 120

Figure 2. The computational result of the PMP example

3.2 The PMP example

Figure 3 illustrates the format implementing a simple CPMP example. In terms of CPMP, the format and settings are the same with the case PMP except that the capacity of each facility (or point) and one more constraint equation "S8:W8 ≤ S9:W9" need to be set. The matrix L which is the products of matrix D and matrix U presents the conditions of loading in each facility. The capacity of the facilities was 100 units.

As can be seen in Figure 4, the cost is 200 under the capacity constrain. The result reveals that the facilities are located at node 1, 3 and 4. The facility located at node 1 services node 1 and 5 and their demands are totally 100 units which satisfied the capacity constrain.

Distance

D	1	2	3	4	5
1	0	5	7	6	4
2	5	0	2	4	6
3	7	2	0	6	8
4	6	4	6	0	2
5	4	6	8	2	0

U	1	2	3	4	5	Total
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

L	1	2	3	4	5
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
Total	0	0	0	0	0
C_j	100	100	100	100	100

Demand

W					
1	60	60	60	60	60
2	20	20	20	20	20
3	50	50	50	50	50
4	70	70	70	70	70
5	40	40	40	40	40

Locations

	1	2	3	4	5	Total
y_j	0	0	0	0	0	0

Name	Range
D	=C3:G7
W	=C11:G15
U	=K3:O7
L	=S3:W7
Cj	=S9:W9

Costs: 0

Cell	Formulas	Copied to
P16	=SUMPRODUCT(D,W,U)	
P3	=SUM(K3:O3)	P4:P7
K11	=K3	
L11	=L4	
M11	=M5	
N11	=N6	
O11	=O7	
P11	= SUM(K11:O11)	
S3	= C11*K3	S3:W7
S8	= SUM(S3:S7)	T8:W8

Solver settings:
Min: P16
By changing: U
Subject to:
U = Binary
P3:P7 = 1
K3:K7 <= K3
L3:L7 <= L4
M3:M7 <= M5
N3:N7 <= N6
O3:N7 <= O7
P11 = p
S8:W8 <= S9:W9

Figure 3. A small example of the CPMP example

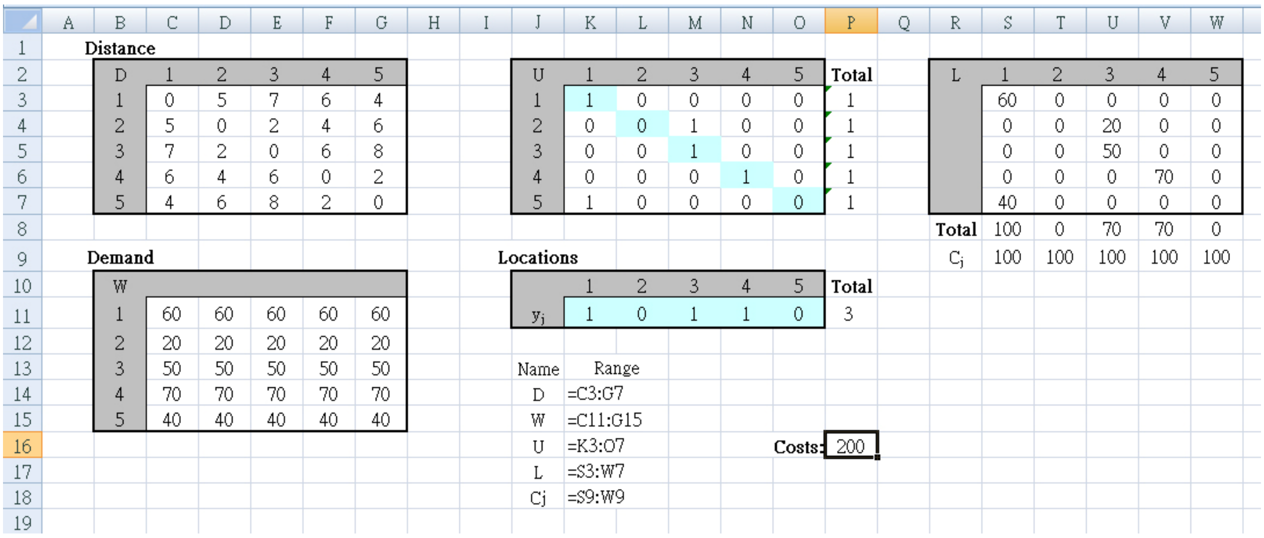


Figure 4. The computational result of the CPMP example

3.3 The MCLP example

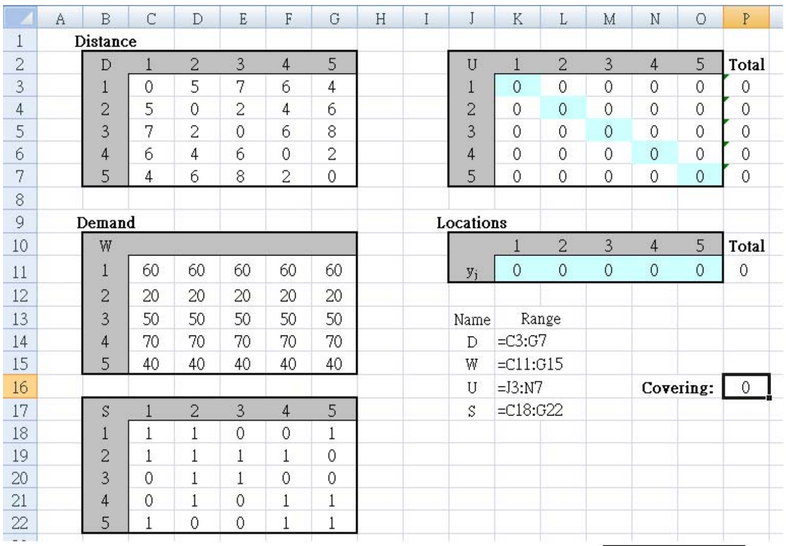
In a similar way, the objective function of the MCLP can be represented as below:

$$\max G = \max \sum_{i=1}^{n} \sum_{j=1}^{n} w_i s_{ij} x_{ij}$$

$$= \text{SUMPRODUCT}(W,\ S,\ U) \qquad (10)$$

Defining the matrix of decision variables $S_{[n \times n]}$ as s_{ij}=1 if the distance between node i and j, d_{ij}, is no longer than the maximum covering distance, d_s; otherwise x_{jj}= 0. The representation of the MCLP and the Solver setting can be seen in Figure 5.

As shown in Figure 6, the facilities are located at node 1, 3 and 5. The node 2 and 4 is covered by the facility at node 1 and 5 respectively. The total covering demands is 240 units.



Cell	Formulas	Copied to
P16	=SUMPRODUCT(W,S,U)	
P3	=SUM(K3:O3)	P4:P7
K11	=K3	
L11	=L4	
M11	=M5	
N11	=N6	
O11	=O7	
P11	= SUM(K11:O11)	
C18	= IF <= 5, 1(True), 0(False)	C18:G22

Solver settings:
Max: P16
By changing: U
Subject to:
U = Binary
P3:P7 = 1
K3:K7 <= K3
L3:L7 <= L4
M3:M7 <= M5
N3:N7 <= N6
O3:N7 <= O7
P11 = p

Figure 5. A small example of the MCLP

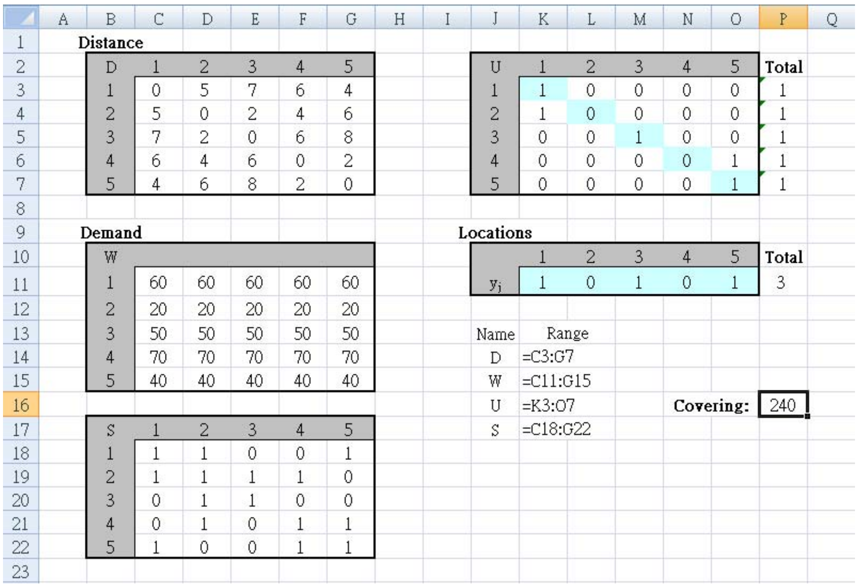


Figure 6. The computational result of the MCLP example

4 Computational Results

All test runs were processed on a laptop computer with an Intel Pentium M processor operating at 1.73GHz. The operating system was Microsoft XP and Excel 2007 was the spreadsheet software. The standard Solver comes bundled with Microsoft Excel. For nonlinear problems, it is limited to problems of up to 200 decision variables and 100 constraints in addition to bounds on the variables. The problems solved far exceed the limits, hence, this study adopted the *Premium Solver Platform V7.0* as the Solver and chose the *large scale LP/Quadratic* as the Solver engine. The *Premium Solver Platform*, a product by Frontline, is an upgrade for the Solver that comes with Microsoft Excel and includes speed and accuracy improvements to the standard Excel Solver. There are several Solver engine provided by *Premium Solver Platform* which can be selected to solve problems, however, none is better than the *large scale LP/Quadratic* for these problems in terms of efficiency. All settings of the Solver were the same as those described in Section 3.

4.1 The experimental results of PMP

Sixteen problems from [23] with n = 100 and n = 150 and various values of p were solved and the results are shown in Tables 1 and 2. The column values of SM are computational results obtained using this approach. In addition, the results were compared with those presented by [15]. They proposed an efficient genetic algorithm (GA) for the PMP, hence, it means that the solutions are not consistent every time. In the instance of n = 100, the average gap of the solutions by Solver is 1.38 %, and the performance is inferior to GA. The process times of these instances (n = 100) are within 30 seconds. In the instance of n = 150, the average gap is 0.89 % and the computational times are no longer than 3 minutes. The solutions using this approach are better than those of GA in the instance where the value of p is large. Although the results show differences in the optimal values, this approach can obtain best solutions for the PMP that the n is small than 55 in seconds.

4.2 The experimental results of CPMP

A set of six problems used by [18] that correspond to real data from the Brazilian city of São José dos Campos were used. Their dimensions (n, p) were (100, 10), (200, 15), (300, 25), (300, 30), (402, 30) and (402, 40), respectively. The computational results of these problems are shown in Table 3. The data of the best known also are from Díaz and Fernández. Most solutions have a difference with the best known and the average gap is 1.08 %. However, the objective value (51508.41) is smaller than the best known (52541.72) in the SJC4b problem. For the problem (n = 402) they were left to run for two hours.

4.3 The experimental results of MCLP

Lorena and Pereira [26] proposed a lagrangean/surrogate (L/S) heuristic for the MCLP. This current work solves some problems, SJC324, SJC402 and SJC500, from their paper and compares the different results (shown in Tables 4, 5 and 6). The number of facilities range from 1 to the minimum needed to obtain full coverage. The service distances, d_s, are designated as 800, 1200 and 1600. The positive gap value means that the solution obtained by this approach is covering more demands than it is by L/S. This approach performs better than L/S in case of (n, d_s) = (402, 1200) and (500, 800). As the problem size grew, the solution time increased dramatically. The CPU times were about 8 hours for the largest problem (n = 500). In addition, Tables 7 and 8 present the solutions of the problems when the p value is bigger and the d_s is short.

5 Discussion

One of the most important discussions in operation planning is facility planning, in which facility location and layout are considered. Selecting a facility location is a very important decision for firms because they are costly and difficult to reverse, and they entail a long term commitment. Furthermore location decisions have an impact on operating costs and revenues. For instance, a poor choice of location might result in excessive transportation costs, lack of qualified labor and supplies of raw materials, lost of competitive advantage, or some condition that would be unfavorable to operations.

In this work a spreadsheet approach for the PMP, CPMP and MCLP was proposed. There are a number of benefits associated with using Excel to solve location problems. Excel is the most widely distributed spreadsheet package in the world. It provides a user-friendly environment for setting up and solving various optimization problems and a robust set of built-in data analysis tools and features that can be used to sort, summarize, and display important information used for decision making. A motivating factor for this study was to provide

Table 1. The compared results of the PMP on the Galvão problem (100 nodes)

					Gap (%)		
No.	*n*	*p*	Optimal	SM	SM	GA(best)	GA(worst)
1	100	5	5703	5869	2.91	0.00	0.00
2	100	10	4426	4518	2.08	0.32	2.76
3	100	15	3893	3968	1.93	0.00	0.23
4	100	20	3565	3622	1.60	0.00	0.25
5	100	25	3291	3335	1.34	0.03	0.15
6	100	30	3032	3066	1.12	0.00	0.07
7	100	35	2784	2813	1.04	0.00	0.18
8	100	40	2542	2570	1.10	0.00	0.12

Gap = 100* (test value – optimal value) / optimal value

Table 2. The compared results of the PMP on the Galvão problem (150 nodes)

					Gap (%)		
No.	*n*	*p*	Optimal	SM	SM	GA(best)	GA(worst)
1	150	5	10839	11027	1.73	0.00	0.62
2	150	15	7390	7491	1.37	0.00	0.77
3	150	20	6454	6569	1.78	0.12	0.67
4	150	25	5875	5996	2.06	0.00	0.70
5	150	35	5192	5197	0.10	0.04	0.37
6	150	45	4636	4637	0.02	0.11	0.26
7	150	50	4374	4377	0.07	0.14	0.21
8	150	60	3873	3873	0.00	0.08	0.21

Table 3. The computational results of the CPMP

Problem	*n*	*p*	Best Known	SM	Gap (%)
SJC1	100	10	17288.99	17288.99	0.00
SJC2	200	15	33270.94	34416.98	3.44
SJC3a	300	25	45338.01	46811.56	3.25
SJC3b	300	30	40635.90	40896.16	0.64
SJC4a	402	30	61928.91	62609.72	1.10
SJC4b	402	40	52541.72	51508.41	-1.97

Gap = 100* (test value – best known value) / best known value

Table 4. The compared results of the MCLP (324 nodes)

n	p	d_s	Demand attended L/S	Demand attended SM	Gap (%)	Coverage (%)
324	1	800	5461	5325	-2.49	43.82
324	2	800	8790	9126	3.82	75.10
324	3	800	11604	11269	-2.89	92.73
324	4	800	12106	12061	-0.37	99.25
324	5	800	12152	12152	0.00	100.00
324	1	1200	9932	9720	-2.13	79.99
324	2	1200	11555	11885	2.86	97.80
324	3	1200	12152	12152	0.00	100.00
324	1	1600	12123	11929	-1.60	98.16
324	2	1600	12152	12152	0.00	100.00

Gap = 100* (test value – L/S value) / L/S value
Coverage = 100* (demand attended value / total demand value)

Table 5. The compared results of the MCLP (402 nodes)

n	p	d_s	Demand attended L/S	Demand attended SM	Gap (%)	Coverage (%)
402	1	800	6555	6555	0.00	41.01
402	2	800	11339	11339	0.00	70.94
402	3	800	14690	14690	0.00	91.90
402	4	800	15658	15658	0.00	97.96
402	5	800	15970	15970	0.00	99.91
402	6	800	15984	15984	0.00	100.00
402	1	1200	10607	12260	15.58	76.70
402	2	1200	14832	15450	4.17	96.66
402	3	1200	15984	15984	0.00	100.00
402	1	1600	15438	15438	0.00	96.58
402	2	1600	15984	15984	0.00	100.00

Table 6. The compared results of the MCLP (500 nodes)

n	p	d_s	Demand attended L/S	Demand attended SM	Gap (%)	Coverage (%)
500	1	800	7944	7944	0.00	40.31
500	2	800	12454	12454	0.00	63.20
500	3	800	15730	15730	0.00	79.82
500	4	800	17794	17851	0.32	90.58
500	5	800	18859	18938	0.42	96.10
500	6	800	19525	19525	0.00	99.08
500	7	800	19692	19692	0.00	99.92
500	8	800	19707	19707	0.00	100.00
500	1	1200	10726	10726	0.00	54.43
500	2	1200	18070	18070	0.00	91.69
500	3	1200	19393	19384	-0.05	98.36
500	4	1200	19707	19707	0.00	100.00
500	1	1600	14804	14804	0.00	75.12
500	2	1600	19668	19668	0.00	99.80
500	2	1600	19707	19707	0.00	100.00

Table 7. The computational results of the MCLP (324 nodes)

n	*p*	d_s	Demand attended	Coverage (%)
324	20	150	7485	61.59
324	30	150	9302	76.55
324	40	150	10485	86.28
324	50	150	11311	93.08
324	60	150	11840	97.43
324	80	150	12152	100.00

Table 8. The computational results of the MCLP (402 nodes)

n	*p*	d_s	Demand attended	Coverage (%)
402	20	150	9007	56.35
402	30	150	11141	69.70
402	40	150	12721	79.59
402	50	150	13985	87.49
402	60	150	14985	93.75
402	80	150	15984	100.00

readers with a methodology that is easy to understand, flexible, and allows them to take advantage of numerous built-in features associated with a readily available software package. The candidate solution generated for each scenario is reevaluated against all of the other randomly generated scenarios in the model. The user can compare and contrast numerous alternatives and view key metrics associated with each candidate

solution – such as average cost, the maximum value of a particular variable across a range of possible scenarios, and the average level of service. Spreadsheets have many advantages, which do not limit themselves to the large number of reprogrammed functions, the power of the graph module or the editing possibilities. Using Excel, these results can easily be presented in both graphical and tabular formats. In addition, a major asset of spreadsheets resides in the speed with which a model may be designed.

When developing professional models with a spreadsheet, one has to cope with the limits of the spreadsheet. In previous versions of Excel, the limits were 256 columns and 65,536 rows. When used with Excel 2007, the *Premium Solver Platform V7.0* supports worksheets with up to 16,384 columns and 1,048,576 rows. The complete model must be held in RAM, which may be a problem if you do not have a machine with enough memory.

The solution search stops if one of several termination conditions as following is met. (1) Declaring "optimality", due to the optimality criteria have been met to within a specified tolerance. (2) Declaring that a default (or user-specified) time or iteration limit has been exceeded. In practice, running solver again from its stopping point took care of this problem. (3) Terminating on "fractional change", which means that the algorithm is making very slow progress - the difference between the objective values at successive points is less than some tolerance for a specified number of consecutive iterations. (4) Declaring that a feasible point cannot be found or that a feasible nonoptimal point has been obtained but a direction of improvement cannot be found. The first of these is a good outcome, indicating location of a local optimum, the second often occurs quite close to a true optimum, and the third is easily corrected by increasing the maximum iterations or time allowed. The fourth outcome may indicate a poorly specified model.

A useful trick to accelerate the search time and improve the quality of solution is to provide Solver with a feasible starting point in that the default initial guess (where all control variables are set to zeros) is not feasible to problems. Spreadsheet solvers require the user to specify starting values for the decision variables. The chosen values will determine which local optimum is reached when the algorithm terminates. Bad starting values can cause an algorithm to fail or to make slow progress. Using other initial values and tuning the solver options may improve the solution. Excel's binomial random number generator was used to assign starting values automatically. It is advisable to try several starting values for the decision variables. When the same solution is produced from each, one can have more confidence that a global minimum is actually obtained. However, this work set the initial values of matrix U to zeros to find the results in section 4 and did not adopt this trick. Theoretically, the trick helps the Solver to find optimal solutions if there is a great deal of test time.

Though solvers interfaced to spreadsheets do not allow the user as much control as standalone or algebraic modeling system solvers, there are a number of options that the user can invoke to aid in solving problems. Among these are options for computing derivatives and for setting various tolerances, which would otherwise be set to defaults by the system. The feasibility tolerance ("precision" option in Excel's solver) controls how accurately a constraint must be satisfied. The fractional change tolerance ("convergence" option in Excel) specifies the amount by which the objective value must differ from (on a relative basis) its previous value in a specified number of iterations in order for the algorithm to continue.

6 Conclusions

This study introduces a spreadsheet method to solve PMP, CPMP and MCLP and provides examples of how spreadsheets can be used to implement this method. The implementation of the matrix formula formulation of the models of these three presented problems makes them easily adapted to spreadsheets. The tool, spreadsheets, is very easy to understand and use, and related data is also easy to update. Computational experiments have been carried out with different specimen data sets and have been compared with other methods in order to evaluate the performance of the approach. The obtained results show that the approach can generate good quality solutions within a reasonable computational time. The solutions of this method are better than those of state of the art approaches for some instances. The mathematical model of other location problems is similar with PMP, CPMP or MCLP, hence, they can also apply this approach to find solutions.

References:

[1] Daskin M, *Network and discrete location: models, algorithms and applications*, New York: Wiley Interscience, 1995. pp. 500.

[2] Brandeau ML, Chiu SS, An overview of representative problems in location research, *Management Science*, Vol.35, No.6, 1989, pp. 645-674.

[3] Pereira MA, Lorena LAN, Senne ELF, A column generation approach for the maximal covering location problem, *International Transactions in Operations Research*, Vol.14, No.4, 2007, pp. 349-364.

[4] Hakimi SL, Optimum distribution of switching centers in a communication network and some related graph theoretic problems, *Operations Research*, Vol.13, 1965, pp. 462-745.

[5] Narula SC, Ogbu UI, Samuelsson HM, An algorithm for the *p*-median problem, *Operations Research*, Vol.25, 1977, pp. 709-713.

[6] Galvão RD, A dual-bounded algorithm for the *p*-median problem, *Operations Research*, Vol.28, 1980, pp. 1112-1121.

[7] Galvão RD, Raggi LA, A method for solving to optimality uncapacitated location problems, *Annals of Operations Research*, Vol.18, 1989, pp. 225-244.

[8] Teitz MB, Bart P, Heuristic methods for estimating the generalized vertex median of a weighted graph, *Operations Research*, Vol.16, 1968, pp. 955-961.

[9] Pizzolato ND, A heuristic for large size *p*-median location problems with application to school location, *Annals of Operations Research*, Vol. 50, 1994, pp. 473-485.

[10] Rosing KE, ReVelle CS, Heuristic concentration: two stage solution construction, *European Journal of Operational Research*, Vol.97, 1997, pp. 75-86.

[11] Rolland E, Schilling DA, Current JR, An efficient tabu search procedure for the *p*-median problem, *European Journal of Operational Research*, Vol.96, 1996, pp. 329-342.

[12] Hansen P, Mladenovic N, Variable neighborhood search for the *p*-median, *Location Science*, Vol.5, 1998, pp. 207-226.

[13] Chiyoshi F, Galvão RD, A statistical analysis of simulated annealing applied to the *p*-median problem, *Annals of Operations Research*, Vol.96, 2000, pp. 61-74.

[14] Chaudhry SS, He S, Chaudhry PE, Solving a class of facility location problems using genetic algorithms, *Expert Systems*, Vol.20, No.2, 2003, pp. 86-91.

[15] Alp O, Erhan E, Drezner Z, An efficient genetic algorithm for the *p*-median problem, *Annals of Operations Research*, Vol.122, 2003, pp. 21-42.

[16] Osman IH, Christofides N, Capacitated clustering problems by hybrid simulated annealing and tabu search, *International Transactions in Operational Research*, Vol.1, 1994, pp. 317-336.

[17] Lorena LAN, Furtado JC, Constructive genetic algorithm for clustering problems, *Evolutionary Computation*, Vol.9, No.3, 2001, pp. 309-327.
[18] Díaz JA, Fernández E, Hybrid scatter search and path relinking for the capacitated *p*-median problem, *European Journal of Operational Research*, Vol.169, 2006, pp. 570-585.
[19] Pirkul H, Gupta R, Rolland E, VisOpt: a visual interactive optimization tool for P-median problems, *Decision Support Systems*, Vol.26, 1999, pp. 209-223.
[20] Reese J, Solution methods for the *p*-median problem: An Annotated Bibliography, *Networks*, Vol.48, 2006, pp. 125-142.
[21] Church RL, ReVelle CS, The maximal covering location problem, *Papers of the Regional Science Association*, Vol.32, 1974, pp. 101-118.
[22] Schilling D, Jayaraman V, Barkhi R, A review of covering problems in facility location, *Location Science*, Vol.1, 1993, pp. 25-55.
[23] Galvão RD, ReVelle CS, A lagrangean heuristic for the maximal covering location problem, *European Journal of Operational Research*, Vol.88, 1996, pp. 114-123.
[24] Dwyer FR, Evans JR, A branch and bound algorithm for the list selection problem in direct mail advertising, *Management Science*, Vol.27, 1981, pp. 658-667.
[25] Downs BT, Camm JD, An exact algorithm for the maximal covering location problem, *Naval Research Logistics Quarterly*, Vol.43, 1996, pp. 435-461.
[26] Lorena LAN, Pereira MA, A lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition, *International Journal of Industrial Engineering*, Vol.9, No.1, 2002, pp. 409-419.
[27] Maric M, Tuba M, Kratica J, Parameter adjustment for genetic algorithm for two-level Hierarchical Covering Location Problem, WSEAS Transactions on Computers, *WSEAS Transactions on Computers*, Vol.7, No.6, 2008, pp. 746-755.
[28] Conway DG, Ragsdale CT, Modeling optimization problems in the unstructured world of spreadsheets, *Omega International Journal of Management Science*, Vol.25, No.3, 1997, pp. 313-322.
[29] Albright SC, *VBA for Modelers: Developing decision support systems with Microsoft Excel*, Pacific Grove, CA: Duxbury Press, 2001.
[30] Rasmussen R, QAP-not so hard in spreadsheets, *Omega International Journal of Management Science*, Vol.35, 2007, pp. 541-552.
[31] Eiselt HA, Sandblom CL, *Decision analysis, location models, and scheduling problems*, New York: Springer, 2004.