World Applied Sciences Journal 13 (12): 2436-2443, 2011
ISSN 1818-4952

Algorithm to Facility Location Problem on a Line

M.E. Berberler, A. Çaliskan, U.G. Nuriyev and B. Sahiner

Department of Mathematics, Faculty of Science, University of Ege, 35100, Bornova, Izmir, Turkey

Abstract: We assume that m facility units (hospital, bank, post office, fire department, police station, bus station, etc.) are requested to be established for n cities which are on the same line. We have to choose m cities where the facility units will be established such that total distance from all cities to nearest facility units is minimum. Here, it is supposed that the person who is in the city which does not have a facility unit will prefer a city which is the nearest among all the other cities having a facility unit. In this problem, we assume that any two cities cannot have the same location. The only selection criterion is the distance between these locations. In this paper, a new mathematical model was proposed and based on this model, a new dynamic programming method was developed and its program was written in DELPHI. The program is available at http://sci.ege.edu.tr/~math/projects/flpsol/.

Key words: p-median . facility location problem . mathematical modeling . dynamic programming

INTRODUCTION

Facility location problem is to locate certain facility units in cities with particular distance on a line. Our purpose is to minimize total distance from the cities without facility units to the cities with facility units. Here, it is supposed that the person who is in the city with a facility unit will prefer to the nearest city among all cities with facility units. Furthermore, any two cities must not be on the same location. Considering that the recommended algorithm will be used to solve real life problem, the algorithm must solve the problem as soon as possible.

At first, one can think that the problem can be solved by brute-force, that is, the combination of (n,m). In this method, set of all possibilities will be found with this combination and the minimum of all these possibilities will be the solution of the problem. The number of elements of the set of all possibilities for n cities and m post office is

$$\binom{n}{m} = \frac{n!}{(n-m)!m!}$$

But this method can not yield to a solution when n and m are large numbers. For example, when n = 250 and m = 10 the number of all possibilities is

$$\binom{250}{10} = 219005316087032475$$

and this is a very large number. It takes *2.31487* years to determine all the possibility on 3 GHZ PC, therefore this is not a realistic method. It is seen that the best method to solve this type of problems is the one that are based on dynamic programming [1-7].

Love (1976) proposed a dynamic programming formulation for one dimensional facility location problem which has nonnegative requirements [2]. Brimberg and Revelle (1998) used linear programming for facility location on a line [3]. Brimberg and Mehrez (2001) solved facility location problem on a line considering weight factors in all cities by using dynamic programming [4]. Brimberg, Korach, Eben-Chaim and Mehrez (2001) found solution for facility location problem with capacitated facilities on a line by using dynamic programming [5]. Dement'ev (1965) discussed a mathematical problem for modeling similar to the above problems and used dynamic programming to solve it in his paper [6]. Gimadutdinov (1969) examined properties of solution which was found by this method [7].

In this paper, we assumed that the capacity of facility units is unlimited and that there is no weight factor (such as population) for all cities. Having considered the distances between cities as the only criteria, we constructed the mathematical model of the problem and presented a fast solution by developing an algorithm based on dynamic programming.

We will keep using the same term for "city", but use the term "post office" for the term "facility unit" in the following sections of this paper. This paper consists of 7 sections: In section 2, the problem is defined; in

Corresponding Author: Dr. Sahiner, B., Department of Mathematics, Faculty of Science, University of Ege, 35100, Bornova, Izmir, Turkey

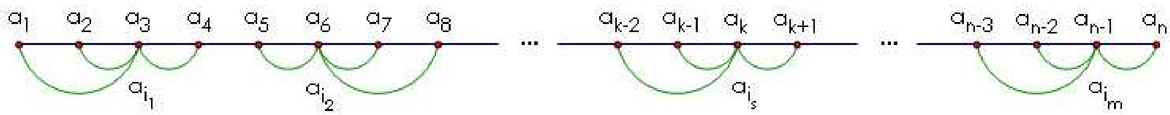

Fig. 1: The location of cities and post offices

section 3, the model of the problem is constructed; in section 4, after the problem is solved for m = 1, the required theorems are given and the way of filling Table 1 which gives solution of the problem in case m>1 is explained; in section 5, the solution algorithm of the problem is offered; in section 6, the result of computation experiments is given; and in section 7 the conclusions of our study are mentioned.

DEFINATION OF THE PROBLEM

Let $A_n = \{a_1, a_2,\ldots,a_n\}$ be the set of cities. We have to choose m cities from n cities in order to locate post offices so that the total distance of all cities to the nearest post office is minimum. We use $P_n^m = \{a_{i_1}, a_{i_2},\ldots,a_{i_m}\}$ to illustrate the selected m cities in order to locate the post offices among n cities (the set A_n)

Let the distance from a_i to a_{i+1} be $d_{i,i+1}$ and the distance from a_{i+1} to a_i be $d_{i+1,i}$. Note that $d_{i,j} = d_{j,i1}$ and $d_{i,i} = 0$, $i = 1,\ldots,n$, $j = 1,\ldots,n$.

MATHEMATICAL MODEL OF THE PROBLEM

The model of the problem can be defined as

$$\min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i\neq j}}^{n} d_{i,j} x_{ij} \qquad (1)$$

subject to:

$$\sum_{j=1}^{n} x_{ij} = 1 - y_i,\ i = 1,\ldots,n, \quad i \neq j \qquad (2)$$

$$0 \leq x_{ij} \leq y_j,\ j = 1,\ldots,n, i = 1,\ldots,n, \quad i \neq j \qquad (3)$$

$$\sum_{j=1}^{n} y_j = m \qquad (4)$$

$$y_j = 0 \vee 1,\ j = 1,\ldots,n \qquad (5)$$

n: The number of cities.
m: The number of post offices.
$d_{i,j}$: The distance from the city i to the city j.
x_{ij}: The decision variable which shows whether the people in the city i goes to the city j or not.
If $x_{ij} = 1$ then the people in city i go to the city j.
If $x_{ij} = 0$ then the people in city i don't go to the city j.
y_j: The decision variable which shows whether a post office will be located in the city j or not.
If $y_j = 1$ then the post office will be located in the city j and a_j will be in the set of post offices P.
If $y_j = 0$ then the post office will not be located in the city j and a_j will not be in the set of post offices P.

In the mathematical model above, expression (1) is the object function which indicates the total distance of the tour that will be done. Expression (2) indicates that a person may go to only once city from each city. Expression (3) controls the selection of post offices. Expression (4) supplies the total number of post offices. Expression (5) determines whether the post office will be located in a city or not.

SOLUTION METHOD OF THE PROBLEM

Investigation of the problem for m = 1: In the set of i number of cities, $L_i^1(a_j)$ indicates the total distance of the roads on the line from all the cities to the city a_j in which the post office is located. In the same set, $R_i^1(a_j)$ indicates the total distance of the roads on the line from the city a_j, in which the post office is located, to the cities after a_j.

$$S_i^1(a_j) = L_i^1(a_j) + R_i^1(a_j),\ i = 1,2,\ldots,n,\ j = 1,2,\ldots,i$$

In the set of i number of cities, P_i^j denotes the set of optimal selections for locating post offices in j number of cities.
Note that

$$d_{i,j} = d_{j,i}, d_{i,i} = 0,\quad i = 1,\ldots,n,\quad j = 1,\ldots,n$$

is used in the following calculations.

It is clear that if a post office is located for 1 city, then

$$P_1^1 = \{a_1\},\ L_1^1(a_1) = 0,\ R_1^1(a_1) = 0,\ S_1^1(a_1) = 0$$

If a post office is located for 2 cities, then $P_2^1 = \{a_1\}$. When the number of optimal solutions is more than one, we will choose the city on the left as a solution. For example, in the situation above, although $P_2^1 = \{a_2\}$ might be a solution, we will choose $P_2^1 = \{a_1\}$.

$$L_2^1(a_1) = 0,\ R_2^1(a_1) = d_{2,1} \Rightarrow S_2^1(a_1) = 0 + d_{1,2} = d_{1,2}$$
$$(L_2^1(a_2) = d_{1,2}, R_2^1(a_2) = 0 \Rightarrow S_2^1(a_2) = d_{1,2} + 0 = d_{1,2} = S_2^1(a_1))$$

If a post office is located for 3 cities, then

$$P_3^1 = \{a_2\} \Rightarrow L_3^1(a_2) = d_{1,2}, R_3^1(a_2) = d_{3,2} \Rightarrow S_3^1(a_2) = d_{1,2} + d_{3,2} = d_{1,2} + d_{2,3}$$

If a post office is located for 4 cities, then

$$P_4^1 = \{a_2\} \Rightarrow L_4^1(a_2) = d_{1,2},\ R_4^1(a_2) = d_{3,2} + (d_{3,2} + d_{4,3}) \Rightarrow S_4^1(a_2) = [d_{1,2}] + [d_{3,2} + (d_{3,2} + d_{4,3})] = d_{1,2} + 2d_{2,3} + d_{3,4}$$

If a post office is located for 5 cities, then

$$P_5^1 = \{a_3\} \Rightarrow L_5^1(a_3) = (d_{1,2} + d_{2,3}) + d_{2,3},\ R_5^1(a_2) = d_{4,3} + (d_{4,3} + d_{5,4}) \Rightarrow$$
$$S_5^1(a_3) = [(d_{1,2} + d_{2,3}) + d_{2,3}] + [d_{4,3} + (d_{4,3} + d_{5,4})] = d_{1,2} + 2d_{2,3} + 2d_{3,4} + d_{4,5}$$

If a post office is located for 6 cities, then

$$P_6^1 = \{a_3\} \Rightarrow L_6^1(a_3) = (d_{1,2} + d_{2,3}) + d_{2,3},\ R_6^1(a_3) = d_{4,3} + (d_{4,3} + d_{5,4}) + (d_{4,3} + d_{5,4} + d_{6,5})$$
$$S_6^1(a_3) = [(d_{1,2} + d_{2,3}) + d_{2,3}] + [d_{4,3} + (d_{4,3} + d_{5,4}) + (d_{4,3} + d_{5,4} + d_{6,5})] = d_{1,2} + 2d_{2,3} + 3d_{3,4} + 2d_{4,5} + d_{5,6}$$

If a post office is located for 7 cities, then

$$P_7^1 = \{a_4\} \Rightarrow L_7^1(a_4) = (d_{1,2} + d_{2,3} + d_{3,4}) + (d_{2,3} + d_{3,4}) + d_{3,4} R_7^1(a_4) = d_{5,4} + (d_{5,4} + d_{6,5}) + (d_{5,4} + d_{6,5} + d_{7,6})$$
$$S_7^1(a_4) = [(d_{1,2} + d_{2,3} + d_{3,4}) + (d_{2,3} + d_{3,4}) + d_{3,4}] + [d_{4,5} + (d_{4,5} + d_{5,6}) + (d_{4,5} + d_{5,6} + d_{6,7})]$$
$$S_7^1(a_4) = d_{1,2} + 2d_{2,3} + 3d_{3,4} + 3d_{4,5} + 2d_{5,6} + d_{6,7}$$

Solution of the problem for m = 1: First, we will give the following theorem and prove it.

Theorem 1:

$$P_q^1 = \left\{ a_{\lceil \frac{q}{2} \rceil} \right\} \qquad (6)$$

where $\lceil x \rceil$ denotes the minimum integer greater than x.

Proof: Having assumed that there are q cities and that one post office will be located in each city, let us calculate the total distances of the roads.

For i = 1

$$P_q^1 = \{a_1\} \Rightarrow L_q^1(a_1) = 0, R_q^1(a_1)$$
$$= d_{2,1} + (d_{2,1} + d_{3,2}) + ... + (d_{2,1} + d_{3,2} + ... + d_{q,q-1})$$
$$S_q^1(a_1) = (q-1)d_{1,2} + (q-2)d_{2,3} + ... + 2d_{q-2,q-1} + d_{q-1,q}$$

For i = 2

$$P_q^1 = \{a_2\} \Rightarrow L_q^1(a_2) = d_{1,2},\ R_q^1(a_2)$$
$$= d_{3,2} + (d_{3,2} + d_{4,3}) + ... + (d_{3,2} + d_{4,3} + ... + d_{q,q-1})$$
$$S_q^1(a_2) = (q-1)d_{1,2} + (q-2)d_{2,3}$$
$$+ (q-3)d_{3,4} + ... + 2d_{q-2,q-1} + d_{q-1,q}$$

For i = 3

$$P_q^1 = \{a_3\} \Rightarrow L_q^1(a_3) = (d_{1,2} + d_{2,3}) + d_{2,3} = d_{1,2} + 2d_{2,3}$$

$$R_q^1(a_3) = d_{4,3} + (d_{4,3} + d_{5,4}) + (d_{4,3} + d_{5,4} + d_{6,5}) + \dots + (d_{4,3} + d_{5,4} + \dots + d_{q,q-1})$$

$$= (q-2)d_{4,3} + (q-3)d_{5,4} + (q-4)d_{6,5} + \dots + 2d_{q-1,q-2} + d_{q,q-1}$$

$$S_q^1(a_3) = d_{1,2} + 2d_{2,3} + (q-2)d_{3,4} + (q-3)d_{4,5} + \dots + 2d_{q-2,q-1} + d_{q-1,q}$$

Therefore;
For i = k

$$S_q(a_k) = [d_{1,2} + 2d_{2,3} + \dots + (k-1)d_{k-1,k}] + [(q-k)d_{k,k+1} + (q-(k-1))d_{k+1,k+2} + \dots + 2d_{q-2,q-1} + d_{q-1,q}]$$

By keeping locating post offices in appropriate cities, we get the following sums, respectively.

1. $(q-1)d_{1,2} + (q-2)d_{2,3} + (q-3)d_{3,4} + (q-4)d_{4,5} + \dots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$
2. $d_{1,2} + (q-2)d_{2,3} + (q-3)d_{3,4} + (q-4)d_{4,5} + \dots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$
3. $d_{1,2} + 2d_{2,3} + (q-3)d_{3,4} + (q-4)d_{4,5} + \dots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$
4. $d_{1,2} + 2d_{2,3} + 3d_{3,4} + (q-4)d_{4,5} + \dots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$

....
....
....

k. $d_{1,2} + 2d_{2,3} + 3d_{3,4} + \dots + (k-1)d_{k-1,k} + (q-k)d_{k,k+1} + \dots + 2d_{q-2,q-1} + d_{q-1,q}$

....
....
....

q-2. $d_{1,2} + 2d_{2,3} + 3d_{3,4} + \dots + (q-3)d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$

q-1. $d_{1,2} + 2d_{2,3} + 3d_{3,4} + \dots + (q-3)d_{q-3,q-2} + (q-2)d_{q-2,q-1} + d_{q-1,q}$

q. $d_{1,2} + 2d_{2,3} + 3d_{3,4} + \dots + (q-3)d_{q-3,q-2} + (q-2)d_{q-2,q-1} + (q-1)d_{q-1,q}$

Here, the difference between sum 1 and sum 2 is (q-2)$d_{1,2}$, the difference between sum 2 and sum 3 is (q-4)$d_{2,3}$, the difference between sum 3 and sum 4 is (q-6)$d_{3,4}$,... and the difference between k^{th} and $(k+1)^{th}$ is

$$d_{k,k+1}[(q-k)-k] = d_{k,k+1}(q-2k)$$

Hence, while k increases, these sums decrease until the sign of expression q-2k changes, but afterwards the sums begin to increase.

As a result, when $k = \lceil q/2 \rceil$, minimum value of the sums is obtained, thus the optimum condition for locating a post office in one city among the *q* number of cities is found. In other words, the k^{th} city is chosen so that a post office can be located in. Thus, the total distance will be,
When q = 2k

$$S_q^1(a_k) = d_{1,2} + 2d_{2,3} + \dots + (k-1)d_{k-1,k} + kd_{k,k+1} + (k-1)d_{k+1,k+2} + \dots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$$

$$= \sum_{i=1}^{k} id_{i,i+1} + \sum_{i=1}^{k-1}(k-i)d_{k+i,k+i+1} = \sum_{i=1}^{k-1} i(d_{i,i+1} + d_{2k-i,2k-i+1}) + kd_{k,k+1} = \sum_{i=1}^{k-1} i(d_{i,i+1} + d_{q-i,q-i+1}) + kd_{k,k+1}$$

when q = 2k+1

$$S_q^1 = d_{1,2} + 2d_{2,3} + \ldots + (k-1)d_{k-1,k} + kd_{k,k+1} + kd_{k+1,k+2} + (k-1)d_{k+2,k+3} + (k-2)d_{k+3,k+4} + \ldots + 3d_{q-3,q-2} + 2d_{q-2,q-1} + d_{q-1,q}$$

$$= \sum_{i=1}^{k} id_{i,i+1} + \sum_{i=1}^{k}(k+1-i)d_{k+i,k+i+1} = \sum_{i=1}^{k-1} i(d_{i,i+1} + d_{2k+1-i,2k+1-i+1}) = \sum_{i=1}^{k-1} i(d_{i,i+1} + d_{q-i,q-i+1})$$

Solution of the problem for m>1: In order to calculate P_i^j and S_i^j by using the following recursive formula for m>1, the table of i = 1,2,…,n, j = 1,2,…,m needs to be organized (Table 1).

$$P_i^j = P_{i-k}^{j-1} \cup \overline{P}_{k,i}^1 \qquad i = 1,\ldots,n \quad , \quad j = 1,\ldots,m \quad , \quad k = 1,2,\ldots,(i-j)+1 \, , \quad j \ge i \tag{7}$$

$$S_i^j = \min_k \left\{ S_{i-k}^{j-1} + \overline{S}_{k,i}^1 \right\} \qquad i = 1,\ldots,n \quad , \quad j = 1,\ldots,m \quad , \quad k = 1,2,\ldots,(i-j)+1 \, , \quad j \ge i \tag{8}$$

where $\overline{P}_{k,i}^1$ indicates only one post office which will be located optimally when the cities in the set of

$$\overline{A}_{k,i} = \{ a_j \mid i-k+1 \le j \le i \}$$

is taken into consideration and $\overline{S}_{k,i}^1$ indicates the total distance from other cities in $\overline{A}_{k,i}$ to the post office in $\overline{P}_{k,i}^1$. As it is seen, the set of $A_i = \{a_1, a_2, \ldots, a_i\}$ is divided into two parts of $\overline{A}_{k,i}$ and $A_{k,i}$ by the k[th] element of a_k. $\overline{A}_{k,i}$ is the complement of $A_{k,i}$.

$$A_i = A_{k,i} \cup \overline{A}_{k,i} = \{a_1, a_2, \ldots, a_k\} \cup \{a_{k+1}, a_{k+2}, \ldots, a_i\}$$

Since 1[st] row of Table 1 corresponds to the case of m = 1 mentioned in the previous section, the 1[st] row of the table is filled by using formula (6) and next rows are filled by using the formulas (7) and (8).

Theorem 2: P_n^m and S_n^m correspond to the optimal solution, in other words, the formulas (6)-(8) give the optimal solution for each i and j.

Proof: For j = 1 and ∀i, Theorem 2 can be proven by Theorem 1.

Table 1: The table of dynamic programming

P\C	1	2	.	.	.	n
1	P_1^1 S_1^1	P_2^1 S_2^1	.	.	.	P_n^1 S_n^1
2		P_2^2 S_2^2	.	.	.	P_n^2 S_n^2
.			.	.	.	.
.				.	.	.
.					.	.
m						P_n^m S_n^m

When j≥2, for each i, as it is also seen in formulas (7) and (8), the appropriate set of A_i is divided into two sets of $A_{k,i}$ and $\overline{A}_{k,i}$ (k=1,2,…,(i-j)+1) for k times. These two sets are the complements of each other. For each of these sets, the optimal solutions exist and the minimum sum of these two sets is chosen. Since only one element is chosen from the 2[nd] set ($\overline{A}_{k,i}$), formula (6) is used and this becomes the optimal solution according to Theorem 1. In the 1[st] set, since the elements (j-1) which were selected in the previous steps are taken into consideration, these cities will also be in the optimal solution set and the sum of them will give us the optimal solution. Hence, since all the possible cases (for (k=1,2,…,(i-j)+1) contain all the possible cases) taken into consideration, the selected solution is the optimal solution.

SOLUTION ALGORITHM OF THE PROBLEM

In order to find the solution of the problem, Table 1 is filled by using the formulas (6)-(8) and in the last cell, P_n^m gives the optimal selection of post offices, while S_n^m gives the minimum total distance.

First of all, the 1[st] row of Table 1 is filled using formula (6) and then considering this result, the next rows are filled using the formulas (7), (8). In formula (8), the formula (6) is used again in order to calculate $\overline{S}_{k,i}^1$.

COMPUTATION EXPERIMENTS

In this section, there are calculations made on 20 sample problems. The sample problems are formed in the following way:

n (the number of cities) is multiplied by 10, the *n* number of cities within the closed interval of [1,10*n] are randomly located. m is chosen randomly in the

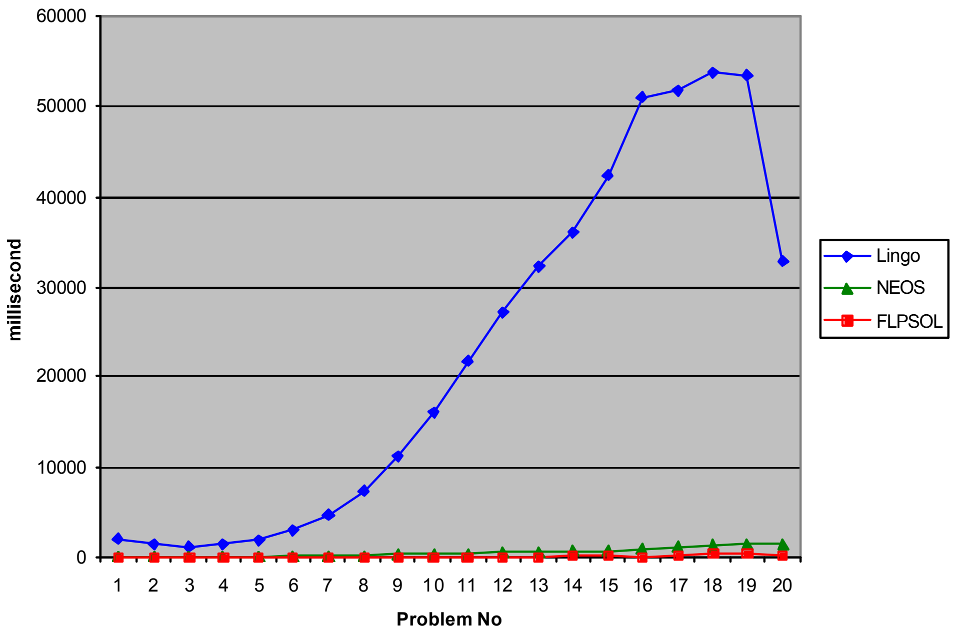

Fig. 2: Lingo, NEOS and FLPSOL

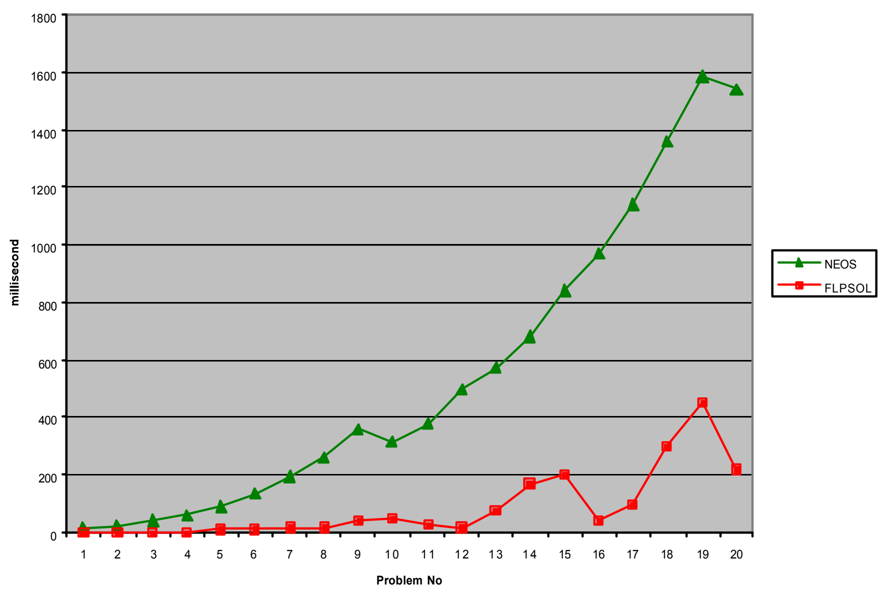

Fig. 3: NEOS and FLPSOL

range [1,n]. The computer program that forms these problems can be found at this page: http://sci.ege.edu.tr/~math/projects/flpsol/.

First, the problems given in section 3 were programmed in Lingo package program [8] and the sample problems are solved with that program. Then, these problems are tried in Cbc solver of NEOS [9]. Finally, these sample problems are solved in our program (FLPSOL). Its results are remarkable. Table 2 shows us the running time of these three programs. For LINGO and FLPSOL, computational experiments are made with PIII 800 MHz, 256 MByte RAM PC. The numbers in the table are milliseconds.

Figure 2 graphically shows us the difference of running time between Lingo, Cbc solver of NEOS and FLPSOL with data which are given in Table 2.

Figure 3 shows the difference in running time between Cbc solver of NEOS and FLPSOL more significantly.

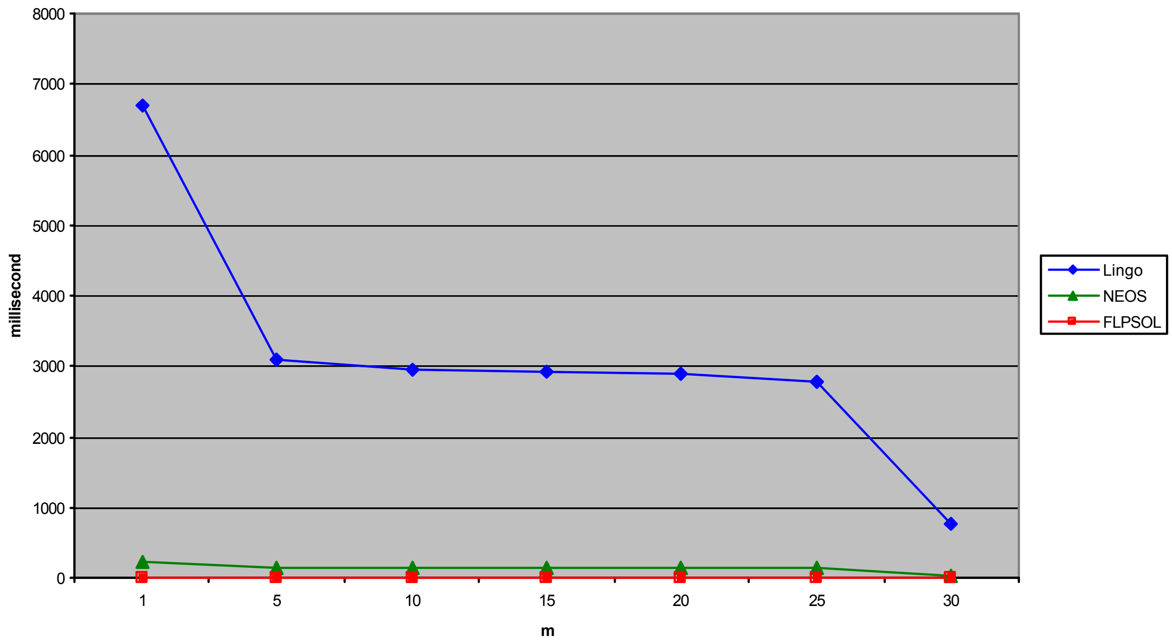

Fig. 4: Behaviors of Lingo, NEOS, FLPSOL with constant number of cities and the varying number of post offices

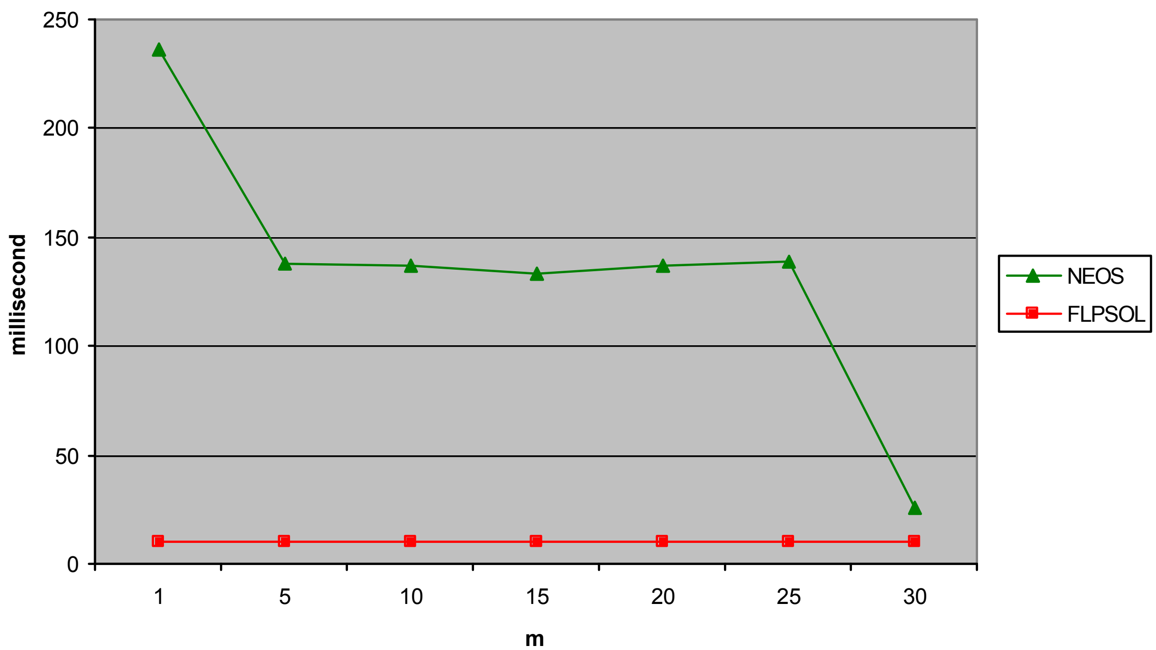

Fig. 5: Behaviors of Lingo, NEOS, FLPSOL with constant number of cities and the varying number of post offices

Furthermore, in this section, by keeping the number of cities constant, the number of the post offices is changed. Then the problem 6 is solved by Lingo, Cbc solver of NEOS and FLPSOL. The running times of these programs are given on Table 3.

Figure 4 graphically demonstrates the behavior of these three programs for problem 6.

Figure 5 shows the difference of running time between Cbc solver of NEOS and FLPSOL more significantly.

Table 2: Running time of three programs to solve sample problems

p	n	m	Opt	Lingo	NEOS	FLPSOL
1	5	4	2	2030	16	0
2	10	2	90	1510	25	0
3	15	11	18	1050	44	0
4	20	14	9	1450	63	0
5	25	17	31	1960	90	10
6	30	10	162	2960	137	10
7	35	28	8	4700	193	20
8	40	15	138	7320	262	20
9	45	32	24	11300	358	40
10	50	30	50	16200	317	50
11	55	13	388	21740	380	30
12	60	10	737	27130	502	20
13	65	28	168	32200	576	80
14	70	58	22	36040	684	170
15	75	56	24	42430	839	200
16	80	8	1505	50920	972	40
17	85	23	502	51670	1143	100
18	90	59	87	53730	1359	301
19	95	81	18	53470	1585	451
20	100	37	348	32890	1543	221

Table 3: The effect of the change in number of the post offices

p	n	m	Opt	Lingo	NEOS	FLPSOL
6_1	30	1	2165	6700	236	10
6_5	30	5	393	3090	138	10
6_10	30	10	162	2960	137	10
6_15	30	15	81	2920	133	10
6_20	30	20	34	2900	137	10
6_25	30	25	9	2780	139	10
6_30	30	30	0	780	26	10

CONCLUSIONS

Our method finds a solution in O(nm) complexity by using dynamic programming method which is suggested to solve one dimensional facility location problem. This situation also shows us that the solution is obtained in the shortest time due to the constraints.

The program of the recommended method is written in DELPHI. And it is available at http://sci.ege.edu.tr/~math /projects/flpsol/.

We want to develop this study in the future by adding the number of passengers criterion to the distance criterion.

REFERENCES

1. Bellman, R. and S. Dreyfus, 1962. Applied Dynamic Programming. The RAND Corporation, R-352, Princeton University Press.
2. Robert F. Love, 1976. One-Dimensional Facility Location Allocation Using Dynamic Programming. Management Science, Vol: 22 (5).
3. Jack Brimberg and Charles Revelle, 1998. Solving The Plant Location Problem On A Line By Linear Programming. Sociedad de Estadistica e Investigacion Operativa, 6 (2): 277-286.
4. Jack Brimberg and Abraham Mehrez, 2001. Location and Sizing of Facilities on a Line. Sociedad de Estadistica e Investigacion Operativa, 9 (2): 271-280.
5. Jack Brimberg, Ephraim Korach, Moshe Eben-Chaim and Abraham Mehrez, 2001. The Capacitated P-Facility Location Problem On The Real Line. Intl. Trans. in Op. Res., 8: 727-738.
6. Dement'ev V.T., 1965. On The Optimal Location Problem of Points in the Range. Discrete Analysis, Novosibirsk. Instute of Mathematics of the Siberian Branch of the Russian Academy of Sciences. (In Russian), 4: 27-34.
7. Gimadutdinov E.Kh., 1969. On The Properties of the Solution to a Problem Regarding Optimal Location of Points in the Range. Control systems, Novosibirsk Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, (In Russian), 2: 77-91.
8. LINGO 11.0-Optimization Modeling Software for Linear, Nonlinear and Integer Programming, www.lindo.com.
9. http://neos.mcs.anl.gov/neos/solvers/milp:Cbc/MPS.html