International Journal of Innovation, Management and Technology, Vol. 1, No. 4, October 2010
ISSN: 2010-0248

Solving Capacitated P-median Problem by Hybrid K-means Clustering and FNS Algorithm

Payman Kaveh, Ali Sabzevari Zadeh and Rashed Sahraeian

***Abstract*— Capacitated p-median problem (CPMP) is one of the popular discrete location problems. CPMP locates p facilities between the candidate sites, in order to satisfy the customers' demands. Usually, in this kind of problems, according to increase of the number of customers and facilities, the solution time of problem will be increased exponentially, so this problem is an NP-hard problem. Therefore, in this paper, we propose a new hybrid algorithm to solve CPMP. In proposed method, k-means clustering algorithm will find a proper solution for Fixed Neighborhood Search (FNS) algorithm. Then, FNS algorithm improves the quality of obtained solutions for standard benchmark instances with facilities locations exchange and omitting the unsuitable candidates' sites. The computational results show the efficiency of the proposed algorithm in regard of the quality of solution.**

***Index Terms*— Capacitated p-median problem, Fixed neighborhood search, k-means clustering.**

I. Introduction

Capacitated *p*-median problem is a special case of capacitated location-allocation problems. In this problem, *p* facilities are located between the candidate sites, in order to satisfy the customers' demands, and of course in a manner that the sum of transportation distances between facilities and customers minimized. We should pay attention: the capacity constraint of each facility should be regarded.

This problem has many applications in real world. Some of its applications are: topological design of computer communications networks (Pirkul [1]), design of a distribution network, where a set of customers are to be supplied by supply points (Fleszar & Hindi [2]), political districting (Bozkaya *et al*. [3]), sales force territories design (Mulvey & Beck [4]).

Whereas, CPMP is an *NP*-hard problem (Garey & Johnson [5]), heuristics methods applied to solve it. One of these methods is column generation approach (Luiz *et al*. [6]). In this method, in each repetition, a set covering location problem is solved. Another method is hybrid Scatter Search (*SS*) and Path Relinking (*PR*) algorithm (Juan & Fernandez [7]). The reasoning considered to use such a method is when path relinking algorithm is used before scatter search method; scatter search algorithm starts to solve the problem with proper initial solution.

Variable Neighborhood Search (*VNS*) is also one of the best methods to solve the capacitated *p*-median problem (Fleszar & Hindi [2]). In this method, for decreasing the solution time, naive lower bound and transshipment lower bound were used for the comparison of new solution and the recent best solution that had ever seen. Also, there is a close kinship between *CPMP* and several other combinatorial optimization problems. If there are no capacity constraints, the problem restricts to the classical *p*-median problem (*PMP*) (Hansen & Mladenovic [8]). In the other hand, if the set of medians are fixed, the problem restricts to the Generalized Assignment Problem (*GAP*) (Yagiura *et al*. [9]).

In this paper, a new hybrid algorithm is proposed to solve *CPMP*. Since, *FNS* is a local search algorithm and it's unable to make the initial solution, *k*-means clustering algorithm is used to generate an initial solution. Then, in *FNS* algorithm, the quality of obtained solutions for standard benchmark instances is improved by proper exchanging the locations of facilities. The rest of the paper is organized as follows: section II introduces the mathematical model of the problem. Section III describes the proposed solution method. Section IV presents the result of computations and their analysis. Finally, Section V concludes with directions for future research.

II. Mathematical Model

Suppose $N=\{1,\ldots,n\}$ and $J=\{1,\ldots,m\}$ show the numbers of customers and candidate sites for locating facilities, respectively. Also, c_{ij}, d_i, b_j show the distance between customer i^{th} ($i\in N$) and facility j^{th} ($j\in J$), the demand of customer i^{th} and the available capacity of located facility in the site j, respectively. Note that, $c_{ii}=0$. The decision variables of problem can be defined as follows:

$y_j=1$, if a facility is located at site $j\in J$, otherwise 0.

$x_{ij}=1$, if customer i^{th} allocates to the facility which is in location j^{th}, otherwise 0.

So, the mathematical model of *CPMP* will be (Juan & Fernandez [7]):

Minimize

$$\sum_{i\in N}\sum_{j\in J} c_{ij}\,x_{ij} \qquad (1)$$

Manuscript received October 3, 2010.

Payman Kaveh is master of Industrial Engineering, Shahed University, Tehran, Iran (e-mail: kaveh_peyman@yahoo.com).

Ali Sabzevari Zadeh is master of Industrial Engineering, Shahed University, Tehran, Iran (e-mail: alisabzevary@gmail.com).

Rashed Sahraeian is assistant professor, Shahed University, Tehran, Iran (email: Sahraeian@shahed.ac.ir).

Subject to

$$\sum_{j\in J} x_{ij} = 1, \ \forall \ i \in N \tag{2}$$

$$\sum_{j\in J} y_j = p \tag{3}$$

$$\sum_{i\in N} d_i \ x_{ij} \le b_j \ y_j, \ \forall \ \ j\in J \tag{4}$$

$$x_{ij} \in \{0,1\}, \ \ y_j \in \{0,1\} \ \forall \ i\in N \ \ , \ \forall \ j \in J \tag{5}$$

In this model, objective function (1) minimizes total distance between facilities and demand points. Constraints (2) ensure that, each customer will be assigned to exactly one facility. Constraint (3) guarantees that the p facilities are located exactly. The constraints (4) ensure that the capacity of every located facility is not violated and constraints (5) impose the binary variables.

III. Solution Method

Whereas, solving the *CPMP* with exact methods is so time-consuming, in this part the new hybrid heuristic method will be proposed to solve the CPMP, which is a combination of *k*-means clustering and *FNS* algorithms.

A. K-means clustering algorithm

K-means clustering algorithm (McQueen [10]) is one of the simplest unsupervised learning algorithms that solve the well known clustering problems. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. The main idea is to define *k* centroids, one for each cluster.

According to above discussion, the steps of *k*-means clustering algorithm can be summarized as follow:

Step 1. Select *k* centroids randomly as initial centroids of *k* clusters (Initial solution).

Step 2. Allocate each customer to the nearest centroids, such that *k* new clusters are created.

Step 3. For each *k* new created clusters, recalculate new centroids.

Step 4. Repeat the steps 2 and 3, so much until new centroids for all clusters in each repetition are fixed.

Note that, values of *k* and *p* are same. Fig.1 shows a simple example of the steps of *k*-means clustering algorithm.

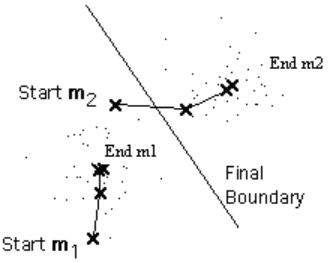

Fig.1. An example for *k*-means clustering steps.

In Fig.1 M_1 and M_2 points are initial centroids. These points were selected randomly. The sign of × shows location of these points. After implementation of *k*-means algorithm steps, *Final Boundary* shows final customers of two clusters. *K*-means clustering algorithm has two shortcomings: first, the *k*-means algorithm does not necessarily find the optimal configuration, corresponding to the global objective function minimum. Second, the algorithm is also significantly sensitive to the initial randomly selected cluster centroids. So, reducing the effect of second shortcoming, *k*-means clustering algorithm will be run multiple times, using the different initial centroids. Since, the above algorithm uses different initial centroids for clustering the customers, this criterion is used to compare the clusters and select best one.

$$Min \ \ Z = \sum_{i=1}^{n}\sum_{j=1}^{k} \left\| X_i^{(j)} - C_j \right\|^2 \tag{6}$$

At above function, $X_i^{(j)}$ shows the location of customer *i* which is allocated by center C_j to the cluster *j*. This function minimizes the sum of squared distance between each customer and centroid of cluster that customer belong to it.

In the proposed hybrid algorithm, the *k*-means clustering algorithm is performed 20 runs with different initial centroids and selects best clustering of customers. Then, a Single Facility Location Problem (*SFLP*) is solved for each of *k* clusters and specifies best facility location between candidate sites. This solution is an initial solution for *FSN* algorithm. It is important to note that, there isn't any capacity constraint for each cluster in *k*-means algorithm.

B. Fixed Neighborhood Search (FNS) Algorithm

In this section, the *FNS* algorithm steps will be described. Firstly, the k^{th} neighborhood of a solution in *FNS* algorithm is defined as follow:

«The k'^{th} neighborhood of a solution contains all the solution that differ from the current one in the location of exactly k' facilities. It means, k' facilities are removed from the current solution and k' new facilities are replaced.»

The steps of *FNS* Algorithm are described as follow:

Step 1. Identify *k'* and *maxitr* values, and then $r \leftarrow 1$.

Step 2. Find an initial solution and call it *S*.

Step 3. Create the k'^{th} neighborhood of the solution of *S*. Call it *Nk'*(*S*).

Step 4. Generate a point *S'* at random from the k'^{th} neighborhood of *S* (*S'* ∈ *Nk'*(*S*)).

Step 5. If *F(S')* < *F(S)*, then $S \leftarrow S'$, $r \leftarrow 1$ and go to the step 3.

Step 6. If *F(S')* > *F(S)*, then $r \leftarrow r+1$, and If $r >$ *maxitr*, then stop algorithm. Otherwise go to the step 4.

C. The modified FNS algorithm

There are three main differences between modified *FNS* algorithm and FNS algorithm. First, in modified *FNS* algorithm, there are few solutions in the k'^{th} neighborhood. Because, some unsuitable candidate sites are eliminated. Second, we use other termination criterion to stop algorithm in modified version of *FNS*. Third, the modified *FNS* algorithm has a tabu list.

1) Two new methods for omitting unsuitable candidate sites

Omitting unsuitable candidate sites has two main advantages. First, the solution time of problem decrease. Second, the proposed method finds the optimal solution with high probability. It is because, unsuitable solutions are not considered. In this section, two new methods are proposed for omitting unsuitable candidate sites.

In the former method, first k' facilities are removed from current solution(s) randomly. Then, for each one of p-k' remained facilities at the current solution, h nearest candidate sites is specified. In the next step, all of specified candidate sites in previous step are omitted. In this method, the value of h will be specified empirically and by trial and error.

In the latter method, there is a rule that according to it, in the most countries, facilities are located in the center and inside area of country, and in the borders area, there aren't much facilities. It is because; customers exist in one side of facilities. So, they can't service many customers with short distances. Hence, by using this reality, some candidate sites are omitted. To do this, a rectangular area is plotted by using candidate sites as follow:

$$(X^{\min}, X^{\max})=(Min(Xj), Max(Xj)) \quad \forall\ j\in J \qquad (7)$$

$$(Y^{\min}, Y^{\max})=(Min(Yj), Max(Yj)) \quad \forall\ j\in J \qquad (8)$$

After plotting the rectangular area, for each line i, i=1,2,3 and 4, N_i, i=1,2,3 and 4, nearest candidate sites are specified to line i. In the next step, these sites are omitted. For example, in Fig.2, the black circles shows candidate sites which were omitted by implementation of second method.

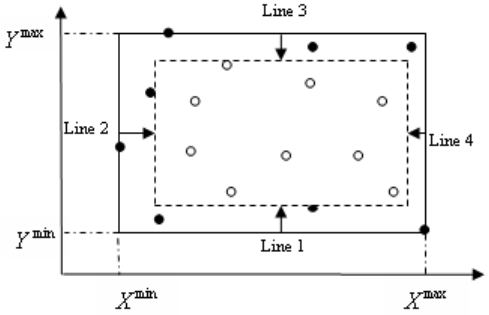

Fig.2. An example for omitting some candidate sites.

Of course, in this method, it is supposed, the coordination of candidate sites and customers are same. Moreover, the value of N_i will be specified empirically and by trial and error.

Now, omitting some unsuitable candidate sites base on two proposed methods and using all remained candidate sites, the $\boldsymbol{k}^{\text{th}}$ neighborhood of current solution is created.

2) Termination criterion

In the proposed hybrid algorithm, in addition to the termination criterion of maximum number of iteration, another criterion is also used. In this new criterion, if for each $\boldsymbol{k'}$ different facilities which are removed from the current solution, the best available solution dose not improves, the algorithm will be finished. In the other word, in this new criterion, all available solutions in the neighborhood are considered. Then, algorithm is stopped. According to above discussion, steps of proposed algorithm are shown in Fig.3.

Also, property of having the memory of Tabu Search algorithm (Glover [11], [12]), is used to avoid the re-evaluation of previously obtained solutions.

The naive lower bound (Fleszar & Hindi [2]), was used to make the solution time of proposed algorithm faster. It means; firstly lower bound of each generated solution is calculated. If the value of the lower bound is more than the value of best available solution, then this solution is not examined and a new solution is generated, otherwise, this solution will be considered.

IV. Computational Results

In order to evaluate the performance of proposed algorithm, it was tested on two sets of standard benchmark problem instances. The first set contains 10 problem instances of size n=50 and p=5 (P1 to P10), and the second set contains 10 problem instances of size n=100 and p=10 (P11 to P20). These two standard benchmark instances are available in the OR library (http://mscmga. ms. ic. ac. uk/info. html) (Beasley [13]). The proposed hybrid algorithm was encoded in MATLAB.7.4.0 (R2007a) and carried out on a Pentium IV with 3.2GHZ and 1GB RAM. Also, the Generalized Assignment Problem (GAP) was solved by the bintprog function of MATLAB software. In all problems, k'=1.

Table I, shows the results of proposed algorithm for 10 runs. In this table, objective functions of optimal solutions are showed in second column and the third column show the number of times out of 10 runs, that the proposed algorithm found an optimal solution. You can see the results of k-means clustering algorithm and the average of the best obtained values at 10 runs of the proposed FNS algorithm in forth and fifth column, respectively. The sixth column also shows the percent of deviation from the best-known solution which is calculated by following formula:

$$dev=\frac{Algorithm\ \ solution-Optimal\ solution}{Optimal\ solution}\times 100 \qquad (9)$$

The two last columns show the solution time of problems when they solved by proposed hybrid algorithm and LINGO8.00 software.

To see if the proposed algorithm is efficient or not, we make a comparison between its results and the results of other algorithms consist of *RR*, *SS* and *PR+SS* algorithms (Juan & Fernandez [7]) and *VNS* algorithm (Fleszar & Hindi [2]). Table II, shows the result of this comparison. Except *VNS* algorithm, other considered algorithms performed in 10 runs.

By comparing the results in table II, we can find, the proposed algorithm has an acceptable performance for all problems with size n=50, P=5. Although, in the proposed hybrid algorithm, the k-means clustering algorithm could just find optimal solution in P_4 and P_5 and in the other problems with size n=50, P=5, the *FNS* algorithm be able to find optimal solution in all runs by proper exchanging and omitting the unsuitable candidate sites. In the first instances set, the deviation of results of proposed algorithm from the best-known solution is zero percent.

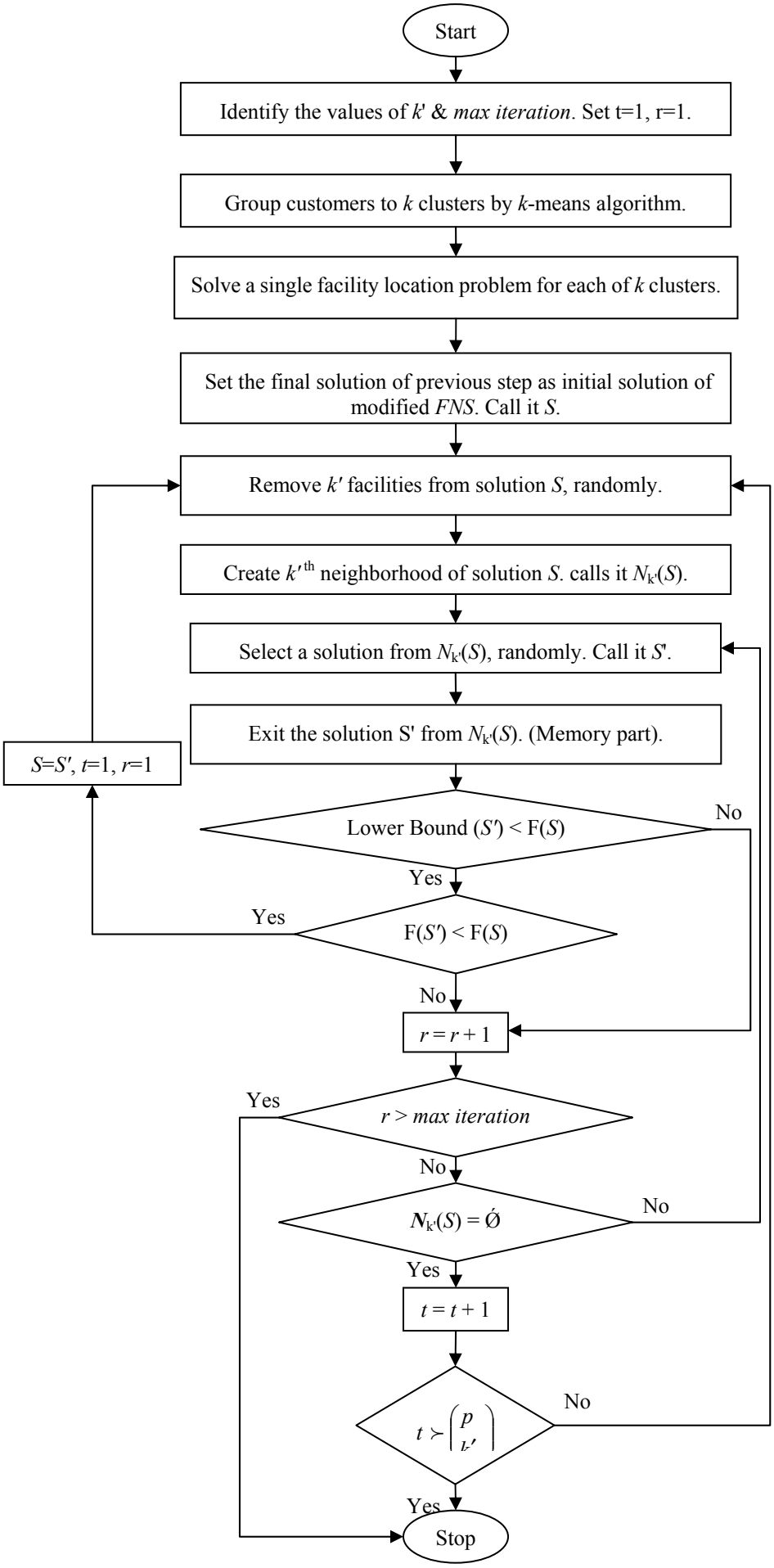

Fig.3. The proposed hybrid algorithm.

TABLE I. PERFORMANCE OF PROPOSED HYBRID ALGORITHM FOR *CPMP*

Problem	Optimal solution	No. opt	Hybrid algorithm		*dev*	CPU time(s)	
			k-means	*FNS*		Hybrid	Lingo8.00
P_1	713	10	714	713	0.00	1.69	18
P_2	740	10	741	740	0.00	1.1	15
P_3	751	10	754	751	0.00	6.2	17
P_4	651	10	651	651	0.00	1.2	15
P_5	664	10	664	664	0.00	3.11	16
P_6	778	10	782	778	0.00	3.66	17
P_7	787	10	847	787	0.00	21.5	74
P_8	820	10	823	820	0.00	348	1010
P_9	715	10	717	715	0.00	3.75	23
P_{10}	829	10	842	829	0.00	73.9	228
P_{11}	1006	10	1025	1006	0.00	130	5700
P_{12}	966	10	984	966	0.00	26	1896
P_{13}	1026	10	1046	1026	0.00	876	942
P_{14}	982	10	996	982	0.00	2400	9480
P_{15}	1091	9	1100	1091.3	**0.02**	1232	6480
P_{16}	954	10	957	954	0.00	7.48	1298
P_{17}	1034	9	1090	1035.4	**0.13**	2603	4020
P_{18}	1043	10	1063	1043	0.00	2172	3540
P_{19}	1031	10	1039	1031	0.00	757	2842
P_{20}	1005	10	1048	1005	0.00	1600	97200

TABLE II. COMPARISON OF PROPOSED ALGORITHM WITH *PR*, *SS*, *PR+SS* AND *VNS* ALGORITHMS FOR *CPMP*

Problem	Optimal solution	Juan & Fernandez [7] , Fleszar & Hindi [2]				Proposed algorithm (*FNS*)	dev				
		PR	*SS*	*PR+SS*	*VNS*		*PR*	*SS*	*PR+SS*	*VNS*	*FNS*
P_1	713	713	713	713	713	713	0. 00	0. 00	0. 00	0. 00	0. 00
P_2	740	740	740	740	740	740	0. 00	0. 00	0. 00	0. 00	0. 00
P_3	751	751	751. 2	751	751	751	0. 00	**0. 03**	0. 00	0. 00	0. 00
P_4	651	651	651	651	651	651	0. 00	0. 00	0. 00	0. 00	0. 00
P_5	664	664	664	664	664	664	0. 00	0. 00	0. 00	0. 00	0. 00
P_6	778	778	778	778	778	778	0. 00	0. 00	0. 00	0. 00	0. 00
P_7	787	787	787	787	787	787	0. 00	0. 00	0. 00	0. 00	0. 00
P_8	820	821. 1	820. 9	820. 9	820	820	**0. 13**	**0. 11**	**0. 11**	0. 00	0. 00
P_9	715	715	715	715	715	715	0. 00	0. 00	0. 00	0. 00	0. 00
P_{10}	829	831. 4	831. 7	831. 4	829	829	**0. 29**	**0. 33**	**0. 29**	0. 00	0. 00
P_{11}	1006	1006.3	1006	1006	1006	1006	**0. 03**	0. 00	0. 00	0. 00	0. 00
P_{12}	966	966	966	966	966	966	0. 00	0. 00	0. 00	0. 00	0. 00
P_{13}	1026	1026	1026	1026	1026	1026	0. 00	0. 00	0. 00	0. 00	0. 00
P_{14}	982	983. 7	984. 2	983. 7	982	982	**0. 17**	**0. 22**	**0. 17**	0. 00	0. 00
P_{15}	1091	1092.8	1093.2	1092. 2	1091	1091.3	**0. 16**	**0. 20**	**0. 11**	0. 00	**0.02**
P_{16}	954	954. 1	954	954	954	954	**0. 01**	0. 00	0. 00	0. 00	0. 00
P_{17}	1034	1034.5	1034	1034	1034	1035.4	**0. 05**	0. 00	0. 00	0. 00	**0.13**
P_{18}	1043	1045	1043	1043. 2	1043	1043	**0. 19**	0. 00	**0. 02**	0. 00	0. 00
P_{19}	1031	1032.1	1032.3	1032	1031	1031	**0. 11**	**0. 13**	**0. 10**	0. 00	0. 00
P_{20}	1005	1008.6	1006.5	1006	1005	1005	**0. 36**	**0. 15**	**0. 10**	0. 00	0. 00

It is noted that, in P_8 and P_{10} problems, *PR*, *SS* and *PR+SS* algorithms are not able to find optimal solution in all 10 runs. Thus, it seems these algorithms have a weaker performance than proposed algorithm, so that in two above-mentioned problems have positive deviations ([%0.13, %0.11, %0.11] and [%0.29, %0.33, %0.29], respectively). Also, in problem P_3, the *SS* algorithm has 0.03 percent deviation. However, *VNS* algorithm finds an optimal solution for each one of P_1 to P_{10} problems. This algorithm has zero percent deviation for all problems.

Obtained results from solving the problems of P_{11} to P_{20} also show that *PR*, *SS* and *PR+SS* algorithms have weaker performance than the proposed algorithm. According to the results of proposed hybrid algorithm for these problems, it could be understood that the new method is able to find a solution in 10 runs (except problem P_{17}), which is better than obtained solutions from *PR*, *SS* and *PR+SS* algorithms. Although, for problem P_{15}, in 10 runs of new method, there is a result which has 0.02 percent deviation, but this deviation is less than the deviation of *PR*, *SS* and *PR+SS* algorithms.

Comparison of the *VNS* and the proposed algorithms results show that, the proposed algorithm had worse results in both P_{15} and P_{17} problems. However, the proposed algorithm finds the optimal solutions for the problems P_{15} and P_{17} in 9 out of 10 runs.

Due to each considered algorithms in tables I and II were encoded in different softwares and different computers were used for performing computational experiments, so we avoid comparing their CPU times. Also, by considering the solution time of the proposed algorithm and Lingo8.00 software, we will find that, the proposed algorithm has better solution times in all different problems.

V. Conclusions And Future Research

In this paper, a new hybrid algorithm is proposed to solve the capacitated *p*-median problem. In this method, *k*-means clustering algorithm provides an initial solution for modified *FNS* algorithm, and then *FNS* algorithm improves the initial solution. We applied two sets of standard benchmark problems instances to evaluate the performance of the proposed algorithm. The results show that the efficiency of the proposed algorithm is due to; first, omitting some unsuitable candidate sites, second, decreasing number of available solutions in the neighborhood and third, avoiding to re-evaluation of previously considered solutions (algorithm's tabu list). We propose some issues for the future research such as, method presentation to choose and remove the *k*' facilities from the solution. This will be more effective in decreasing the number of available solution in the neighborhood and avoiding of local optimal solutions.

References

[1] Pirkul H., "Efficient algorithm for the capacitated concentrator location problem", Computers and Operations Research14 (3), Pages 197-208, 1987.

[2] Fleszar K., K.S. Hindi, "An effective VNS for the capacitated p-median problem", European Journal of Operational Research 191, pages 612–622, 2008.

[3] Bozkaya B., Erkut E., Laporte G., "A tabu search heuristic and adaptive memory procedure for political districting", European Journal of Operational Research 144(1), pages 12-26, 2003.

[4] Mulvey J.M., Beck M.P., "Solving capacitated clustering problems", European Journal of Operational Research 18, pages 317-336, 1984.

[5] Garey M. R, Johnson D. S, "A Guide to the Theory of NP- Completeness", W. H. Freeman and CO., 1979.

[6] Luiz A.N. Lorena, Edson L.F. Senne, "A column generation approach to capacitated p-median problems", Computers & Operations Research 31, Pages 863–876, 2004.

[7] Diaz Juan A, Elena Fernandez, "Hybrid scatter search and path relinking for the capacitated p-median problem", European Journal of Operational Research 169, pages 570–585, 2006.

[8] Hansen Pierre, Mladenovic Nenad, "Variable Neighborhood search for the p-median", Location Science5, pages 207-226, 1997.

[9] Yagiura M., Ibaraki T., Glover F., "A path relinking approach with enjection chains for the generalized assignment problem", European Journal of Operational Research 169(2), Pages 548-569, 2006.

[10] McQueen J.B., "Some methods for classification and analysis of multivariate observations", In: Proceedings of the Symposium on Mathematics and Probability, 5th, Berkely, Vol. 1, AD 669871. University of California Press, Berkeley, CA, Pages, 281-297, 1967.

[11] Glover F., "Tabu search" – part 1, ORSA j. comput. 1. Pages 190-206, 1989.

[12] Glover F., "Tabu search" – part 2, ORSA j. comput. 2. Pages 4-32, 1990.

[13] Beasley J.E., OR – library: "Distributing test problems by electronic mail", Journal of the Operational Research Society41 (11), pages 1069-1072, 1990.