Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 4, No. 1, February 2013), Pages: 89-100
www.jacr.iausari.ac.ir

Solving the Capacitated Clustering Problem by a Combined Meta-Heuristic Algorithm

Narges Mahmoodi Darani[1]✉, **Vahid Ahmadi**[2], **Zahra Saadati Eskandari**[3], **Majid Yousefikhoshbakht**[4]

(1) Department of Mathematic, Malayer Branch, Islamic Azad University, Malayer, Iran
(2) Department of Mathematic, Ahvaz Branch, Islamic Azad University, Ahvaz, Iran
(3) Young Researchers & Elite Club, Fereydan Branch, Islamic Azad University, Fereydan, Iran
(4) Young Researchers & Elite Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran

saadati.zahra@yahoo.com; khoshbakht@iauh.ac.ir

Received: 2012/08/24; Accepted: 2012/10/07

Abstract

The capacitated clustering problem (CCP) is one of the most important combinational optimization problems that nowadays has many real applications in industrial and service problems. In the CCP, a given n nodes with known demands must be partitioned into k distinct clusters in which each cluster is detailed by a node acting as a cluster center of this cluster. The objective is to minimize the sum of distances from all cluster centers to all other nodes in their cluster, such that the sum of the corresponding node weights does not exceed a fixed capacity and every node is allocated to exactly one cluster. This paper presents a hybrid three-phase meta-heuristic algorithm (HTMA) including sweep algorithm (SA), ant colony optimization (ACO) and two local searches for the CCP. At the first step, a feasible solution of CCP is produced by the SA, and at the second step, the ACO, insert and swap moves are used to improve solutions. Extensive computational tests on standard instances from the literature confirm the effectiveness of the presented approach compared to other meta-heuristic algorithms.

Keywords: *Capacitated Clustering Problem, NP-hard Problems, Ant Colony Optimization, Sweep Algorithm, Local Search*

1. Introduction

One of the important parameters in production and services is decreasing the product’s expense; since lower product costs yield to the company’s competitive benefit growth in terms of production and services, the company’s profit increases remarkably. Furthermore, minimizing the delivery cost is one of the cost reduction methods in which the goods are transported from the depot to other places with minimum cost. The cost of delivery can be significantly reduced by using a software management without compromising the services provided to the requesters. The requesters are grouped based on their demands with optimal number of clusters and the minimum cost of each service delivery. Clustering is a difficult combinatorial problem, and clustering algorithm can be either hierarchical or partitioned. Hierarchical algorithm discovers successive clusters using previously established clusters, whereas partitioned algorithm defines all clusters at once.

The capacitated clustering problem (CCP) is one of the most challenging ones in operation research problems, and has been widely studied for many years mainly because of its application in real world problems. Several variations of CCP have been considered in the literature including a capacitated centered clustering problem (CCCP) [1] as well as a capacitated p-median problem (CPMP) [2]. The CCP also provides solution to vehicle routing problem (VRP) [3, 4], and waste collection [5] which involve clustering. Clustering is an unsupervised classification of patterns into groups. The CCP partitions a set of n node into k disjoint clusters with known uniform capacity. During clustering the items with shortest assigning paths from centroids are grouped together. The summation of grouped items should not exceed the capacity of the cluster. Clustering techniques have always been considered as essential tools to solve many practical problems. Previous studies on the CCP are mostly applied to facility location problems and they often focus on the development of solution algorithms. An example of a solution consisting of a set of clusters constructed for a CCP is presented in Figure1.

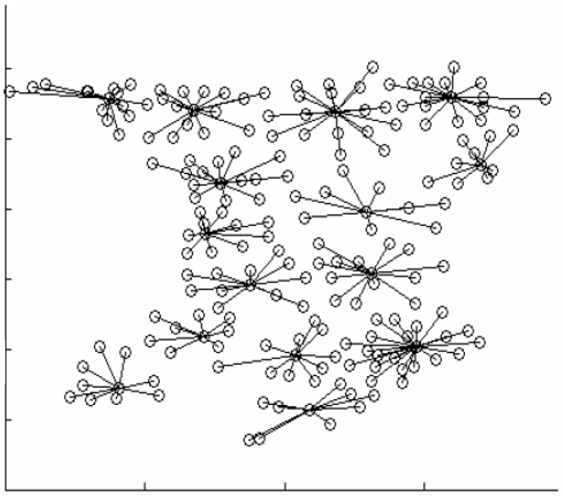

Figure 1. A solution of the CCP

Since it was proven that CCP is a non-deterministic polynomial time (NP-hard) problem, in this paper, a hybrid three-phase meta-heuristic algorithm (HTMA) is developed to solve this problem effectively and efficiently. This algorithm consists of two key phases: construction and enhancement. In the construction phase, a sweep approach (SA) is applied to construct a feasible solution of CCP and in the enhancement phase, ant colony optimization (ACO) and two-stage local searches including the insert and the swap algorithms are used to improve the solution quality with respect to the similarity measurement. The effectiveness of the HTMA is demonstrated using standard data sets and its results compared with state-of-the-art approaches in the literature. The experimental results show that the HTMA confirms the effectiveness of the presented approach.

The remainder of this paper is organized as follows: Section 2, Literature Survey of CCP; Section 3, the proposed framework designed to solve the CPP; Section 4, computational experiments using real standard datasets as well as the effectiveness of the HTMA in comparison to the existing methods; and finally, the paper is concluded in Section 5.

2. Literature Survey

In previous studies a large number of techniques have been offered to solve the different versions of CCP. Generally speaking, these techniques can be classified into exact, heuristic and meta-heuristic methods. As an example, a column generation with a specialized branching technique was proposed by Mehrotra and Trick in 1988 [6], and solved a maximum weighted cluster problem (MWCP) in the sub problem. Baldacci et al. also offered a new exact algorithm by modeling the capacity location problem as a set partitioning problem with cluster-feasibility constraints in 2002 [7]. A combination of the column generation and Lagrangean relaxation techniques was also proposed by Lorena and Senne to solve capacitated *p*-median problems [8]. Recently, Ceselli et al. proposed a computational framework based on column generation and branch-and-price approaches for solving the capacitated network problems in 2009 [9].

Although exact algorithms are more appropriate for small size instances, they are not often suitable for real instances owing to the computational time required to obtain an optimal solution. Besides, the CCP has been proved to be a NP-hard problem. This means that the CCP solution time grows exponentially with the growth in distribution points. In other words, when the problem size increases, the exact methods cannot solve it proficiently, so heuristic and meta-heuristic algorithms are used to solve the problem and settle for the suboptimal answers. Hence, these approaches are used when finding an optimal solution in a reasonable amount of time is not possible for instances with large size. Therefore, a large number of heuristic approaches have been recently developed due to the computational complexity of real-life CCPs. For example, Mulvey and Beck proposed classical sub-gradient heuristics and used randomly generated seeds as an initial solution for solving the CCP in 1984 [10] while in 1992 Koskosidis and Powell extended the work of [10] by an iterative algorithm [11]. This proposed algorithm was more effective than other heuristic algorithms and avoided the specification of seed nodes required by other algorithms. Moreover, the iterative algorithms use self-correcting scheme in three phases including greedy assignment, seed relocation and local exchange. These are not only heuristic algorithms applied to the CCP and the cluster search [12], GRASP-based algorithms [13], and additional heuristics [14, 15] are other examples of heuristic algorithms.

Although heuristic methods solve the CCP in acceptable time, they become trapped in local optimum as shown in Figure. 2 [16], and cannot gain a good suboptimal solution. As a result, in the last 30 years, a new kind of heuristic algorithm called meta-heuristic has developed, which basically tries to combine basic heuristic methods into higher level frameworks aimed at efficiently and effectively exploring a search space. Meta-heuristic algorithms such as ACO, tabu search, particle swarm optimization, simulated annealing, genetic algorithm and others have much better ability than heuristic methods for solving combinatorial optimization problems. For example, Thangiah and Gubbi used a ’cluster first and route second’ for solving VRP [17]. In this algorithm, a genetic algorithm was proposed to find a good cluster of nodes at the first stage. Another genetic algorithm was also offered by How-Ming Shieh and May for solving CCP [18]. In this algorithm, binary coded strings are applied to represent the chromosome, which eliminates the occurrence of infeasible solution. Furthermore, the chromosome is separated into two parts for representing the nodes and the seeds of the clusters. Furthermore, a k-means algorithm which performs correct clustering without pre-assigning the exact number of clusters was introduced by Zalik [19]. França et al. [20] settled a new adaptive tabu search approach to solve the CCP. They used two neighborhood generation mechanisms of the local search heuristic including pairwise interchange and insertion. It should be noted that an adaptive penalty function is used to handle the capacity constraint that improves the convergence and solution quality.

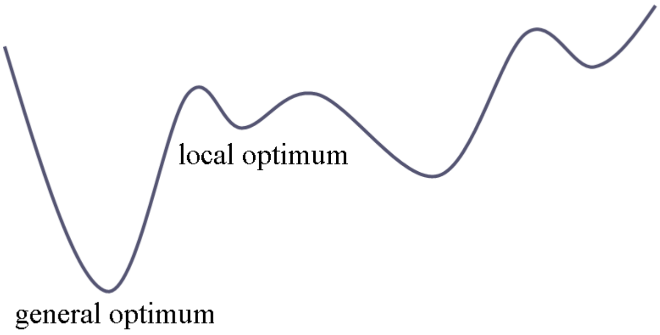

Figure 2. Local and general optimums

3. The Proposed Algorithm

This study proposes a hybrid meta-heuristic algorithm to solve the CPP which can be explained as the problem of designing optimal clusters for a number of nodes subject to side constraints. In this section, first SA and ACO is presented and then the HTMA will be considered in more detail.

3.1 Sweep Algorithm

The SA is one of the famous and efficient heuristic algorithm proposed by Gillett and Miller in 1974 [21]. In this algorithm, the center is the depot and all the nodes are ranked (as shown in formula below) by using their polar coordinate (An(i)= arctan{(y(i)-y(0))/(x(i)-x(0))} in which $-\pi < An(i) < 0$ if $y(i) - y(0) < 0$ and $0 \leq An(i) \leq \pi$ if $y(i) - y(0) \geq 0$. Then, sweeping counterclockwise is started from a node *i*, which has not been visited, from the smallest angle to the larger angle until all nodes are included in a tour.

$$An(1) \leq An(2) \leq ... \leq An(n)$$

3.2 Ant Colony Optimization

Nowadays, the meta-heuristic algorithm is one of the most famous group algorithms for solving combinatorial optimization problems. These algorithms have been developed according to artificial intelligence, biological evolution and/or physics phenomenon, etc. The ACO [22], Genetic Algorithms (GA) [23], Tabu Search (TS) [24], Imperialist Competitive Algorithm [25] and so on [26-27] are several more applied meta-heuristic used for many NP-hard problems in recent years. Among these meta-heuristic approaches, ACO is a new distributed meta-heuristic which simulates the ant's food-hunting behavior. This is a probabilistic technique used for solving problems that do not have a known efficient algorithm. Nonetheless the abilities of a single ant are limited, ants can together find the shortest route between a food source and their nest by using the chemical substances called pheromones. Whenever an ant moves in order to find food, it lays down a trail of pheromone as a communication medium for ants. Therefore, the probability that any ant chooses one path over other increases and the more ants use a given trail, the more attractive it becomes to other ants. On the other hand, the pheromone evaporates over time, so that it will convert less obvious on longer trails, which takes more time to traverse. Consequently, longer trails will be less attractive, which is valuable to the other ants (see Figure. 3 [28]).

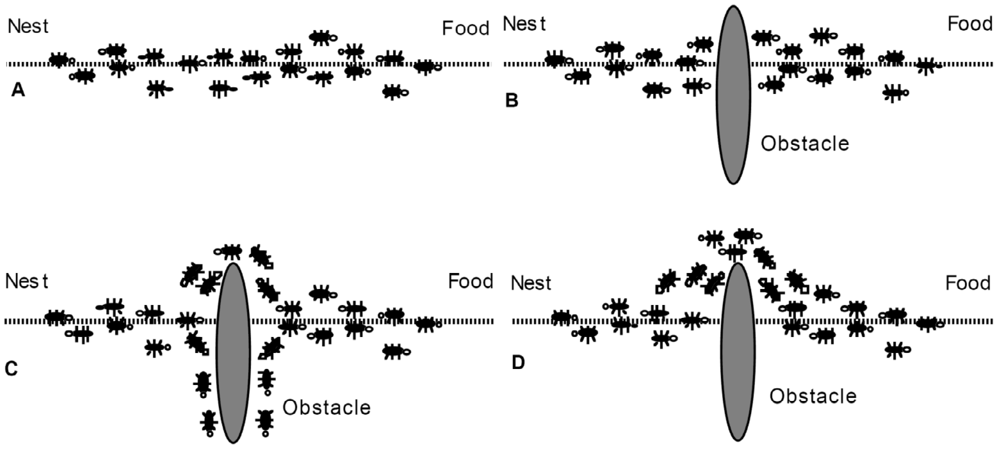

Figure 3. The behavior of ant's food-hunting

The first version of ACO was proposed by Marco Dorigo in 1992 [29]. This algorithm is called ant system (AS) aimed at searching for an optimal path between two nodes in a graph. In order to solve a problem by AS, the considered problem is divided into some sub-problems in which the simulated ants are predictable to choose the next node based on the amount of the pheromone in a trail (τ_{ij}) and the inverse distance to the next node (η_{ij}). In formula (1), the decision to select the unvisited member of *N_i* node by ant *k* located at node *i* is done. Both α and β in this formula are parameters and can be changed by the user.

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum\limits_{j\in N_i} \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & if \quad j \in N_i \\ 0 & if \quad j \notin N_i \end{cases} \tag{1}$$

When ants move from current node *i* to new node *j*, they release pheromone information ($\Delta\tau_{ij}$) on the respective path. Formula (2) shows how to calculate the τ_{ij}.

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

Besides, the algorithm uses pheromone evaporation to prevent quick convergence of ants to a sub-optimal local optimum. In other words, pheromone density is automatically reduced in each iteration by coefficient $(1-\rho)$ in the formula (3). In this, $0 \le \rho \le 1$ is set by the user and is updated at each iteration and τ is the matrix for the existing pheromone on the edges of the respective graph.

$$\tau \leftarrow (1-\rho)\tau \qquad \rho \in [0,1] \tag{3}$$

ACO has a number of versions such as an elite ant system (EAS), ACS, rank based ant system (RAS) and max-min ant system (MMAS); every version having its own advantages and weaknesses. The AS could not produce satisfactory results compared with meta-heuristic algorithms of the time. Therefore, these several variants have been derived from the basic AS and able to produce better results in the past years. These algorithms have also been successfully applied to many combinatorial optimization problems, such as traveling salesman problem (TSP) [30], the multi-depot multiple traveling salesmen problems [31], scheduling problems [32], and vehicle routing problems (VRPs) [33]. In these applications, ACO has been comparative performance to another state of the other meta-heuristics.

3.3 Hybrid three-phase meta-heuristic algorithm

The HTMA consists of four algorithms; including sweep algorithm (SA) as a constructive algorithm, ACO, insert exchange (IE) and swap exchange (SE) as three improvement algorithms. It is noted that a construction algorithm produces a feasible solution while an improvement algorithm can improve the solution produced by a constructive algorithm. In order to improve every cluster in the HTMA, ACO is used while the nodes of each cluster may not be changed. Besides, the insert and swap exchanges are applied for changing nodes between clusters. Here, the details of the HTMA will be analyzed and its steps are described as follows.

3.3.1. Production of a feasible solution

The nodes visited at the first stage of the HTMA are prepared with respect to the depot by sweep algorithm, and then they are set in the array. For this array, the algorithm starts to move from the depot and trip the nodes on the array in the described order until it is not possible to add a further node any more. In other words, if the load of a cluster is more than its capacity, the HTMA returns to the depot and the cluster will be built. This process is repeated until there is not any node to be allocated to the clusters. When all of the clusters are obtained, the center of each cluster should be obtained. For this goal, all the nodes are tested as center and their function objective will be gotten. After that, the best solution and clustering will be chosen as a primitive feasible solution.

3.3.2. Improvement of the feasible solution

After producing a feasible solution by mentioning algorithm; in this stage the ACO, IE and SE algorithms are used to improve the primitive solution. As is shown in Figure 4, in inserting algorithm a node is moved to another route. However, in swap algorithm a node in a certain route is swapped with another node from a different route (see Figure 5). It should be noted that the new solution will be only accepted in state that first, CCP constraints are not violated specially about each cluster’s capacity and second, novel solution will gain better value for problem than previous solutions.

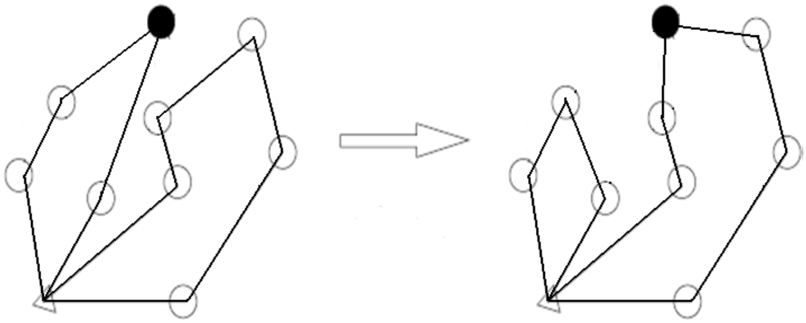

Figure 4. Insert exchange

For achieving this goal, *p* ants are set on the all of the centers of clusters (each ant is set on each cluster) and for each iteration, allocated nodes of each cluster are selected by this method. For example if *i* is a center cluster, the node *j* belonged to this center is chosen by formula (1). After that, all of the ant returned to their center of nodes and selected again other nodes in clusters according to capacity constraints. When the new solution is produced, the insert and swap algorithms are applied for improving the quality of the solution even more.

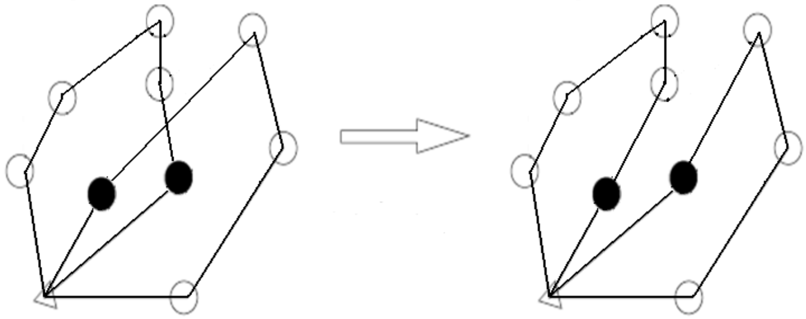

Figure 5. Swap exchange

3.3.3. Local search

Local search is a key part of meta-heuristic algorithms for solving NP-Hard problems and the vast amount of literature show that a promising approach to obtaining high-quality solutions are to couple a local search algorithm with a mechanism to generate initial solutions. Local search can be used on problems that can be formulated as finding a solution maximizing or minimizing of the problem among a number of candidate solutions. A local search approach starts with an initial solution and searches within neighborhoods in the search space for better solutions. If these algorithms find an improved solution in comparison with previous iterations, they move from the current solution to the new solution. These apply local changes are continued, until a solution deemed optimal is found or a time bound is elapsed. Here in the HTMA, a haphazardly nearest neighborhood (HNN) is used to local search. In this algorithm, the center is changed for each cluster. For example if *k* is a center of the cluster *i*, the probability of selecting *j* as a new center is calculated by the formula (4). This formula leads to allocating a new node to center of the cluster with better objective function (lower cost in CCP). In this formula, c_{ij} is the distance between node *i* and node *j*.

$$k_j = \frac{1/c_{ij}}{\sum_{i=1}^{r} 1/c_{ij}} \qquad j = 1,\dots,r \tag{4}$$

3.3.4. Stop conditions

At this stage, the stop condition is checked. If this condition is met, the algorithm ends. Otherwise, if stop condition is not satisfied, the algorithm is iterated again. The stop condition in the HTMA is that the algorithm is iterated *n* (number of nodes) times to end the loop. The best obtained result up to this stage is considered as the best values. If it is not happened, the algorithm goes to step 3.3.2. The pseudo code of HTMA for solving the CCP is presented in Figure 6.

```
1) Producing a feasible solution by the sweep algorithm.
2) Finding center of every cluster by testing all the statuses of cluster's nodes.
3) Setting one ant for each  cluster center.
4) Choosing nodes of the each cluster by ACO.
5) Changing nodes of clusters by the insert and swap algorithms
6) Changing center of each clusters by using the HNN
```

Figure 6. Pseudo code of the HTMA

4. Discussion and Computational Results

In this section, firstly the benchmark problems is introduced and then the results obtained from calculations by the HTMA are compared with other algorithms. These algorithms were executed on two sets of standard benchmark problem instances containing between 50 and 100 nodes of the CCP problem including P1, P2, …, P20. In more detailed examples, 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). The proposed HTMA presented in Section 3, was coded in Matlab 7. All the experiments were implemented on a PC with Pentium 4 at 2.4GHZ and 2GB RAM running Windows XP Home Basic Operating system. Because the HTMA is meta-heuristic, the algorithm is performed for 10 runs and the results are shown in Table 1.

The specifications of these twenty problems and the results of the HTMA over the benchmark instances compared to the results of other meta-heuristic are reported in Table 1. Euclidean distances are used in the all problems. This table gives the following information: the number of nodes (n); the number of clusters (p); the number of independent runs of the HTMA (Runs); the *RR*, *SS* and *PR+SS* algorithms proposed by Juan & Fernandez [34]; the VNS algorithm proposed by Fleszar & Hindi [35], the FNS proposed by Kaveh et al. [36] and the HTMA. Moreover, to show the method's performance more clearly, the best known solutions (BKS) are presented that have been published in the related literature in this table. The combinations of optimal parameters that are obtained from several time tests are shown in formula (5). It should be noted here that except VNS algorithm, other considered algorithms were performed in 10 runs. Furthermore, for each considered algorithms in this table different soft wares were encoded and different computers were used to perform computational experiments, so comparing their CPU times are not considered.

$$\alpha = 1, \beta = 2, \rho = 0.1 \tag{5}$$

By comparing the results of this table, the HTMA has a satisfactory performance for all problems with size *n*=50 and *P*=5. Although the PR, SS and PR+SS algorithms could not find the optimal solution in P8 and P10 with size *n*=50, *P*=5 in all 10 runs, the proposed algorithm is able to find the optimal solution in all these problems. Thus, these algorithms have a weaker performance than PA. It is noted that in the two mentioned problems these algorithms have positive deviations [%0.13, %0.11, %0.11] and [%0.29, %0.33, %0.29], respectively. Furthermore, VNS and the HTMA find an optimal solution for all of *P1* to *P10* problems and they have a zero percent deviation for all problems. Therefore, in the first instances set, the percentage deviation (PD) of the results of the proposed algorithm from the BKS is zero percent in which the PD of the HTMA solution cost comparison to the BKS costs was shown in the formula (6).

$$PD = 100 \times \frac{\left(value\ of\ HTMA\ \text{-}\ value\ of\ BKS\right)}{value\ of\ BKS} \tag{6}$$

Table 1. Comparison results for problem set

Instance	N	p	Runs	PR		SS		PR+SS		VNS		FNS		HTMA		BKS
				Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	
P 1	50	5	10	713	0.03	713	0.03	713	0.03	713	0.17	713	1.69	713	5.12	713
P 2	50	5	10	740	0.02	740	0.02	740	0.02	740	0.05	740	1.1	740	7.89	740
P 3	50	5	10	751	0.26	751. 2	0.26	751	0.27	751	0.19	751	6.2	751	8.11	751
P 4	50	5	10	651	0.04	651	0.04	651	0.04	651	0.11	651	1.2	651	6.31	651
P 5	50	5	10	664	0.18	664	0.18	664	0.18	664	0.27	664	3.11	664	9.21	664
P 6	50	5	10	778	0.09	778	0.09	778	0.09	778	0.11	778	3.66	778	8.87	778
P 7	50	5	10	787	0.13	787	0.13	787	0.13	787	0.31	787	21.5	787	7.89	787
P 8	50	5	10	821. 1	0.12	820. 9	0.20	820. 9	0.21	820	0.92	820	348	820	9.91	820
P 9	50	5	10	715	0.08	715	0.09	715	0.09	715	0.13	715	3.75	715	10.21	715
P 10	50	5	10	831. 4	0.28	831. 7	0.27	831. 4	0.28	829	0.75	829	73.9	829	9.46	829
P 11	100	10	10	1006.3	2.34	1006	3.08	1006	3.20	1006	7.91	1006	130	1006	32.65	1006
P 12	100	10	10	966	0.44	966	0.44	966	0.45	966	4.81	966	26	966	31.47	966
P 13	100	10	10	1026	0.77	1026	0.76	1026	0.77	1026	2.17	1026	876	1026	29.84	1026
P 14	100	10	10	983. 7	3.09	984. 2	1.46	983. 7	3.11	982	10.33	982	2400	982	35.67	982
P 15	100	10	10	1092.8	5.34	1093.2	4.84	1092.2	6.41	1091	10.23	1091.3	1232	1091.3	37.66	1091
P 16	100	10	10	954. 1	3.47	954	4.01	954	4.10	954	4.20	954	7.48	954	41.27	954
P 17	100	10	10	1034.5	2.27	1034	3.07	1034	2.93	1034	5.50	1035.4	2603	1034	45.62	1034
P 18	100	10	10	1045	1.45	1043	6.28	1043. 2	6.29	1043	9.06	1043	2172	1043.2	39.73	1043
P 19	100	10	10	1032.1	4.31	1032.3	4.03	1032	4.42	1031	8.64	1031	757	1031	37.11	1031
P 20	100	10	10	1008.6	4.34	1006.5	5.76	1006	5.49	1005	27.34	1005	1600	1005	51.28	1005

In the second set which contains 10 problem instances of size n=100 and p=10 (P11 to P20), the Obtained results show that the HTMA has better performance than *PR*, *SS* and *PR*+*SS* algorithms. In more details, the proposed algorithm is able to find optimal solution in 10 runs except problem P18 which is better than obtained solutions from *PR*, *SS* and *PR*+*SS* algorithms. On the other hand, Comparison of the *VNS* and the HTMA results show that, the proposed algorithm has almost equal results except in both P15 and P18 problems.

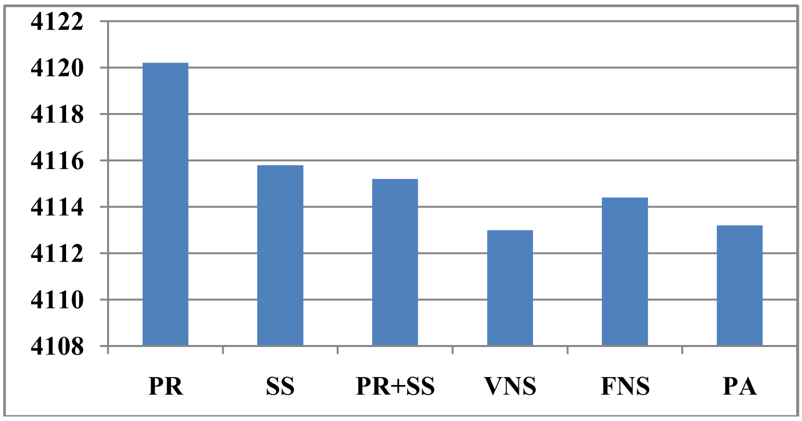

Figure 7. Comparison of sum results for 20 instances between meta-heuristic algorithms

In Figure 7, the sum of the results of each algorithm's solution is reported. From this Figure, it is concluded that the HTMA is one of the best algorithms because this algorithm has found the best solutions for 18 examples out of 20 and has shown a better performance than any other algorithms specially PR, SS and PR+SS. As a result, the order of algorithms in terms of their performance from the worst to the best is: PR, SS, PP+SS, FNS, HTMA and VNS.

5. Conclusions

In this paper, a hybrid algorithm was proposed for solving the CCP which was more efficient than any other meta-heuristic algorithms; especially for large-size problems. In this method, the sweep algorithm provides an initial solution for the HTMA, and then ACO and local search algorithms improve the initial solution. Several standard benchmark instances are used to evaluate the performance of the HTMA. The experimental results show that the HTMA confirms the effectiveness of the presented approach. Some issues for the future research can be proposed here such as, combination of the proposed algorithm and tabu search will yield better results as well as using this algorithm for other versions of the CCP. This will be more effective in improving available solutions and avoiding local optimal solutions.

6. Acknowledgement

The authors would like to acknowledge the Malayer Branch, Islamic Azad University for the financial support of this work.

7. References

[1] M. Negreiros, and A. Palhano, "The capacitated centered clustering problem," *Computers and Operations Research,* 2006.

[2] A.A. Chaves, F. de Assis Correa, and L.A.N. Lorena, "Clustering search heuristic for the capacitated p-median problem," *Advances in Software Computing Series,* 2007.

[3] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, "An exact algorithm for the capacitated vehicle routing problem based on a two commodity network flow formulation," *Operations Research,* 2004.

[4] T.K. Ralphs, "Parallel branch and cut for capacitated vehicle routing," *Parallel Computing,* 2003.

[5] B. Kim, S. Kim, and S. Sahoo, “Waste collection vehicle routing problem with time windows,” *Computers and Operations Research,* 2006.

[6] A. Mehrotra, and M. A. Trick, “Cliques and clustering: A combinatorial approach,” *Operations Research Letters,* 1998.

[7] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, and A. Mingozzi, “A new method for solving capacitated location problems based on a set partitioning approach,” *Computers & Operations Research*, 2002.

[8] L.A. Lorena, and E.L. Senne, “A column generation approach to capacitated p-median problems,” *Computers & Operations Research,* 2004.

[9] A. Ceselli, F. Liberatore, and G. Righini, “A computational evaluation of a general branch-and-price framework for capacitated network location problems,” *Annals of Operations Research*, 2009.

[10] J. Mulvey, and M.P. Beck, “Solving capacitated clustering problems,” *European Journal of Operational Research,* 1984.

[11] I. Koskosidis, and W.B. Powell, “Clustering algorithms for consolidation of node orders into vehicle shipments,” *Transportation Research Part B*, 1992.

[12] A.A., Chaves, and L.A., Lorena, “Clustering search algorithm for the capacitated centred clustering problem”, *Computers & Operations Research*, 2010.

[13] Y. Deng, and J. Bard, “A reactive grasp with path relinking for capacitated clustering,” *Journal of Heuristics*, 2010.

[14] M. Negreiros, and A. Palhano, “The capacitated centred clustering problem,” Computers *& Operations Research*, 2006.

[15] P. Avella, M. Boccia, A. Sforza, and I. Vasil’ev, “An effective heuristic for large-scale capacitated facility location problems,” *Journal of Heuristics*, 2009.

[16] M. YousefiKhoshbakht, and M. Sedighpour, “A Combination of Sweep Algorithm and Elite Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem”, *Proceedings of the Romanian Academy A*, 2012.

[17] S.R. Thangiah, and A.V. Gubbi, “Effect of genetic sectoring on vehicle routing problems with time windows,” *IEEE International Conference on Developing and Managine Intelligent System Projects,* 1993.

[18] H. M. Shieh, and M. D. May, “Solving the capacitated clustering problem with genetic algorithms,” *Journal of the Chinese Institute of Industrial Engineers*, 2001.

[19] K. R. Zalik, “An efficient k-means clustering algorithm,” *Pattern Recognition Letters,* 2008.

[20] P. M. Franca, N. M.Sosa, and V. Pureza, “An adaptive tabu search algorithm for the capacitated clustering problem,” *International Transactions in Operational Research*, 2006.

[21] B.E. Gillett, and L.R. Miller, “A heuristic algorithm for the vehicle dispatch problem,” *Operations Research*, 1974.

[22] M. YousefiKhoshbakht, F. Didehvar, and F. Rahmati, “Modification of the Elite Ant System in Order to Avoid Local Optimum Points in the Traveling Salesman Problem”, *Wulfenia Journal*, 2012.

[23] M. Sedighpour, M. YousefiKhoshbakht, and N. Mahmoodi Darani, “An Effective Genetic Algorithm for Solving the Multiple Traveling Salesman Problem”, *Journal of Optimization in Industrial Engineering*, 2011.

[24] Z. Saadati Eskandari, M. YousefiKhoshbakht, “Solving the Vehicle Routing Problem by an Effective Reactive Bone Route Algorithm”, *Transportation Research Journal*, 2012.

[25] M. YousefiKhoshbakht, M. Sedighpour, “A New Imperialist Competitive Algorithm to Solve the Traveling Salesman Problem,” *International Journal of Computer Mathematics, 2013.*

[26] M. AHMADVAND, M. YOUSEFIKHOSHBAKHT, AND N. MAHMOODI DARANI, “SOLVING THE TRAVELING SALESMAN PROBLEM BY AN EFFICIENT HYBRID METAHEURISTIC ALGORITHM,” *JOURNAL OF ADVANCES IN COMPUTER RESEARCH,* 2012.

[27] M. YOUSEFIKHOSHBAKHT, F. DIDEHVAR, AND F. RAHMATI, “OPTIMIZATION OF VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY USING A MODIFIED TABU SEARCH,” *JOURNAL OF ADVANCES IN COMPUTER RESEARCH,* 2013.

[28] M. YousefiKhoshbakht, M. Sedighpour, and E. Mahmoodabadi, "A Modified Elite ACO Based Avoiding Premature Convergence for Traveling Salesman Problem", *Journal of Industrial Engineering International*, 2011.

[29] M. Dorigo, and L., Ganbardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evolutionary Computing*, 1997.

[30] S.M. Chen, and C.Y. Chien, "Parallelized genetic ant colony systems for solving the traveling salesman problem", *Expert Systems with Applications*, 2011.

[31] S. Ghafurian, and N. Javadian, "An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems," *Applied Soft Computing*, 2011.

[32] D. Merkle, and f. Middendor, "Ant colony optimization with global pheromone evaluation for scheduling a single machine," *Applied Intelligence*, 2003.

[33] X. Zhang, and L. Tang, "A new hybrid ant colony optimization algorithm for the vehicle routing problem," *Pattern Recognition Letters*, 2009.

[34] D. Juan, and E. Fernandez, "Hybrid scatter search and path relinking for the capacitated p-median problem", *European Journal of Operational Research*, 2006.

[35] K. Fleszar, and K.S. Hindi, "An effective VNS for the capacitated p-median problem", *European Journal of Operational Research*, 2008.

[36] P. Kaveh, A. Sabzevari Zadeh and R. Sahraeian, "Solving Capacitated P-median Problem by Hybrid K-means Clustering and FNS Algorithm," *International Journal of Innovation, Management and Technology*, 2010.