A Simulated Annealing Algorithm within the Variable Neighbourhood Search Framework to Solve the Capacitated Facility Location-Allocation Problem

Ragheb Rahmaniani[a,*], Abdolsalam Ghaderi [b], Mohammad Saidi-Mehrabad [c]

[a] MSc, Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

[b] Assistant Professor, Department of Industrial Engineering, University of Kurdistan, P.C. 66177-15177, Sanandaj, Iran

[c] Professor, Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Received 2 September, 2011; Revised 19January, 2012; Accepted 27 February, 2012

Abstract

In this study, we discuss the capacitated facility location-allocation problem with uncertain parameters in which the uncertainty is characterized by given finite numbers of scenarios. In this model, the objective function minimizes the total expected costs of transportation and opening facilities subject to the robustness constraint. To tackle the problem efficiently and effectively, an efficient hybrid solution algorithm based on several meta-heuristics and an exact algorithm is put forward. This algorithm generates neighborhoods by combining the main concepts of variable neighborhood search, simulated annealing, and tabu search and finds the local optima by using an algorithm that uses an exact method in its framework. Finally, to test the algorithms' performance, we apply numerical experiments on both randomly generated and standard test problems. Computational experiments show that our algorithm is more effective and efficient in term of CPU time and solutions quality in comparison with CPLEX solver.
Keywords:Capacitated Facility Location-allocationProblem, Single Allocation, Uncertainty, Hybrid Algorithm.

1. Introduction

Facility location problems are generally concerned with how to locate some facilities and how to allocate a given set of clients to them, subject to various constraints and aiming to optimize some objectives such as minimizing transportation cost. These problems have been extensively investigated and encompass a wide range of literature and numerous applications in operations management (Owen and Daskin, 1998; Melkote and Daskin, 2001; Marianov and Serra, 2002; Manzour-al-Ajdad; Torabi et al., 2012).In line with this subject, many models have been introduced and solved in the literature(Brandeau and Chiu, 1989; Ghiani, Guerriero et al., 2002; Pasandideh and Chambari, 2010; Ghaderi, Jabalameli et al., 2011) in which the Capacitated Facility Location Problem, CFLP, due to its real assumption has attracted the attention of several researchers(Küükdeniz, Baray et al., 2012; Yin and Mu, 2012). The CFL P assumes that each facility has a limited capacity to serve the customers. Likewise, it covers a substantial scope of application, such as determining the location of schools, hospitals, fire service stations; location of warehouses in a supply chain; production planning; telecommunication network design, power stations, and so forth. Although this problem can be easily understood, it is intractable from thecomputational point of view. The CFLP is an NP-hard problem which is generalized from the simple plant location problem. As a result, a lot of solution algorithms have been used to solve the CFLP. In this regard, Lagrangean Relaxation(LR)is widely considered in the literature as an efficient solution algorithm to solve the capacitated problems.Cornuejols, Sridharan et al. (1991) provided an excellent theoretical analysis of all possible Lagrangean relaxations and the linear programming relaxation for the CFLP. Beasley (1993)presented a solution framework based on langrangian to solve different facility location problems. In the proposed method for the CFLP, allocation constraints and facilities' capacity constraints are incorporated into the objective function by Lagrangian multipliers. Barahona and Chudak (2005)firstly provided the linear programming relaxation of the CFLP and then applied the Lagrangean relaxation to solve the linear problem. Similarly, exact methods are developed to solve the CFLP, for example,Sa (1969)relaxed the CFLP to obtain the transportation problem

*Corresponding author E-mail address: Ragheb.rahmaniani@gmail.com

andNauss (1978) relaxed capacities constraints and added a set of surrogate constraints to obtain tighter bounds.

Moreover,Sun (2011) applied the Tabu Search (TS)to solve the CFLP and compared it with the Lagrangean and the surrogate/Lagrangean heuristic methods.In the study, a long term memory based on primogenitary linked the quad tree to store the visited solutions and prohibit them from being generated again.Korupoluet al. (2000) proposed a simple local search heuristic for the capacitated facility location problem. Hindi and Pienkosz (Hindi and Pienkosz, 1999) developed a heuristic based on the Lagrangean relaxation with the VNS to solve the capacitated single source location problem. Fleszar and Hindi (2008) proposed an efficient variable neighborhood search heuristic for the capacitated P-median problem.Manzour-al-Ajdadet al. (2012)developed an iterative two-phase heuristic algorithm to solve the single-source capacitated multi-facility weber problem. They also proposed a Simulated Annealing (SA) algorithm to complete the first phase of their algorithm. Last but not least,Leitner and Raidl (2012) proposed algorithms based on the variable neighborhood search and the greedy randomized adaptive search to solve the connected facility location problem.

Despite these attentions, the researchers have mainly considered the CFLP with deterministic data while in the reality this assumption is rarely satisfied. Facility location decisions are long-term decisions and almost impossible to change. On the other hand, utilized data in managerial decisions are usually incomplete and there is no historical information to estimate the models' key parameters effectively. Therefore, optimizing under uncertainty in location problems has received increasing attention during the last decade(Snyder, 2006; Nikoofal and Sadjadi, 2010; Murali, Ordóñez et al., 2012), and several approaches are proposed to tackle the uncertainty. These approaches typically employ robust or stochastic optimization.Robust models typically minimize the worst scenario which yields in intractable models and too conservative solutions. On the other hand, stochastic models usually minimize the expected cost which may perform poorly in the long run and under a certain realization of random data. However, a new optimization approach is introduced by Snyder and Daskin (2006)which minimizes the expected costs while relative regret in each scenario must not be greater than a positive value, known as the robustness coefficient.They combined the minimum-expected-cost and p-robustness measures together to introduce the stochastic p-robust optimizing model .This approach was adopted for the classical models of the UFLP and P-median problem(PMP). The researchers intended to find a solution that had the minimum-expected-cost while the obtained solution was p-robust, i.e., the cost under each scenario for each feasible solution must be within 100(1+p)% of the optimal cost for that scenario(p is the robustness measurement).They solved the proposed models by using Lagrangian decomposition. The proposed algorithm reduces problems to the multiple-choice knapsack problem. They also discussed a mechanism for detecting infeasibility. In the present study, we use this approach to resolve the uncertainty which may appear in the key parameters of the CFLP. For further discussion on this approaches, the readers may read the work presented by Rahmanini et al.

In short, our problem is locating several capacitated facilities in a network to service a given set of customers. These customers can only receive service from one facility and each customer should be completely satisfied. Moreover, the objective function minimizes transportation and operating costs. In order to face the today's fierce and changing business environment more appropriately, we assume that the key parameters are uncertain and the uncertainty is associated with demands and distances. The uncertain parameters are characterized with a given number of scenarios in which each scenario has a specific probability to occur. Furthermore, we discuss both robust and stochastic formulation and present the stochastic p-robust formulation for the problem in hand. Moreover, from a computational point of view it is very challenging to solve this problem. As a result, we propose an algorithm to tackle the problem effectively. This algorithm is a combination of the variable neighborhood search, simulated annealing, tabu search, and an exact method. To test its performance, we solve a wide range of test instances and compare the computational results of the algorithm with the CPLEX.

The reminder of this paper is organized as follows: in the next section the mathematical formulation of the problem is presented. A solution approach based on the VNS is presented to solve the problem in Section 3.Our numerical experiment is summarized in Section 4 and finally our conclusion is discussed in Section 5.

2. Model Formulation

In this part, we address the problem descriptions and its mathematical formulation. To begin with, we present the notations and assumptions and then we discuss the problem's mathematical formulation.

2.1 Notations and Assumptions

For the sake of convenience in presenting the mathematical formulation in this article, the following notations are defined to be used throughout the paper.The index sets and model parameters are respectively described in Table 1.

Table1
Index sets

Symbol	*Indexed by*	*Description*
I	$i \in \{1, 2, ..., \lvert I \rvert\}$	set of costumers;
S	$s \in \{1, 2, ..., \lvert S \rvert\}$	Set of scenarios;
J	$j \in \{1, 2, ..., \lvert J \rvert\}$	set of potential facility site;

In this problem, we have |I| centers of customers and a number of facilities which are to be located at |J| potential sites. These facilities which have limited capacity, C, should satisfy customers' demands in such a way that the total construction costs and travel costs are minimized. Since locating a large number of facilities is highly undesirable, we confine the number of facilities to V. Moreover, in order to deal with the today's fierce and changing environment more appropriately, we assume that the model's parameters are under uncertainty and we have no specific probability distribution at hand to estimate them. Accordingly, we estimate the demands, set-up costs, and distance by using several discrete scenarios. Each scenario has a specific probability of occurrence, q_S , which also shows the importance of that scenario in decision making.

Table 2
Input parameters

Symbol	*Description*
D_i^s	Demand at node *i* under scenario *s*
f_i^s	The set-up cost of facility on node *j* under scenario *s*
l_{ij}^s	Travel distance between nodes *i* and *j* under scenario *s*
q_s	Probability of scenario *s*
V	Number of maximum allowable facilities for locating
C	Capacity of facilities
p	Desired robustness coefficient
Z_s^*	optimal objective value of the CFLP under the data from scenario *s*

The decisions of the problem in hand include two sets of decisions regarding facilities' location and costumers' assignment. Likewise, according to the stochastic nature of the problem, we partition these decisions into two stages. As a result, the location variables (y) unlike the assignment variables (X) are independent of the s index to reflect the two-stage nature of the problem.

$$y_j = \begin{cases} 1 & \text{if one facility is located at node } j \\ 0 & \text{Otherwise} \end{cases}$$

$$X_{i,j}^s = \begin{cases} 1 & \text{if node } i \text{ is assigned to the facility located at node } j \text{ under scenario } s \\ 0 & \text{Otherwise} \end{cases}$$

2.2 The mathematical formulation of the CFLAP under uncertainty

In the robust optimization approaches, the objective function typically minimizes the worst case or the maximum relative regret over all possible scenarios. If we translate this perspective into a mathematical formulation, the objective function of the problem in hand would be as follows.

$$Min\{max(\sum_{j\in J} f_j^s y_j + \sum_{i\in I}\sum_{j\in J} D_i^s l_{ij}^s X_{ij}^s)\};\ \forall s \in S \quad (1)$$

Robust models due to their mini max structure are hard to solve. Moreover, decisions of a robust model are unsurprisingly too conservative and may impose unnecessary costs in the long run. However, this approach is widely used in the literature to resolve uncertainties especially whenever scenarios' occurrence probabilities are not given. Another alternative approach is stochastic programming models. In general, the typical stochastic programming may yield inexpensive solutions in the long run but perform poorly under certain realizations of the random data. In the stochastic approach, the objective function of the problem by multiplying the probability occurrence of each scenario to the objective function of that scenario would be as follows:

$$Min\{\sum_{s\in S} q_s\,(\sum_{j\in J} f_j^s y_j + \sum_{i\in I}\sum_{j\in J} D_i^s l_{ij}^s X_{ij}^s)\} \quad (2)$$

In Eq.(2),the total expected costs of facilities construction and transportation are minimized.Moreover, each client should be completely satisfied. That means, the sum of total satisfied demand for each customer i, should be equal to its demand. Since we assume that the problem is a single-allocation problem, we have constraint (3).

$$\sum_{j\in J} X_{i\,j}^s = 1 \qquad \forall i \in I \quad (3)$$

In addition, our facilities have a major constraint on their capacity and cannot supply an unlimited amount of demand to the customers and logically the customers should be only allocated to the facilities. By translating these perspectives into mathematical symbols, Eq.(4) represents the capacity constraint of our model.

$$\sum_{j\in J} D_i^s X_{i\,j}^s \le C\, y_j \qquad \forall i \in I,\ \forall s \in S \quad (4)$$

As mentioned before in this section, stochastic models may not perform well under certain realizations of data. For example, when disruption happens, the facilities may fail to serve customers. To overcome this problem and avoid the intractable structure of robust models, Snyder and Daskin (2006) introduced the stochastic p-robust optimization approach. In this approach, the objective function minimizes expected costs while the p-robustness condition is incorporated into constraints. One of the main goals of this approach is to design a more robust system with a little increase in the expected costs.

$$\sum_{j\in J} f_j^s y_j + \sum_{i\in I}\sum_{j\in J} D_i^s l_{ij}^s X_{ij}^s \le (1+p) Z_s^*\ \forall s \in S \quad (5)$$

Accordingly, Eq.(5) represents the p-robustness constraint. In this constraint, the objection function of each scenario($s \in S$) should not be greater than$(1+p)$% optimal cost of that scenario.Finally, Eq.(6) specifies the maximum number of facilities that can be open and Eq.(7 and 8) indicate decision variables.

$$\sum_{j \in J} y_j \leq V \quad (6)$$

$$X_{ij}^s \in \{0, 1\} \forall i \in I, \forall j \in J, \forall s \in S \quad (7)$$

$$y_j \in \{0, 1\} \, \forall j \in J \quad (8)$$

It is also interesting to note that if $p = \infty$ and $|S| = 1$ the presented mathematical formulation would become equivalent to the traditional CFLP.

3. A Fix-and-Optimize Algorithm

The developed model is an NP-Hard model since it is an extension of another NP-Hard problem under uncertainty. Therefore, for realistically sized instances it is very challenging to solve most of the test instances within a reasonable computer CPU time using the known optimization solvers. Therefore, in this article, a fix-and-optimize heuristic procedure which uses an exact method and several meta-heuristics is developed to solve the model.The main procedure of neighborhood generation is based on the Variable Neighborhood Search (VNS)(Brimberg and Mladenovi´c, 1996; Hansen, Mladenovi´c et al., 2010), Simulated Annealing (SA)(Pahlavani and Saidi-Mehrabad, 2011), and Tabu Search (TS)(Sun 2011) while the local search is based on a method that uses branch & bound in its framework. Generally, the VNS generates next neighborhoods based on a specific neighborhood structure (i.e. k). In other words, it randomly selects k basic nodes and replaces them with k non-basic nodes. This algorithms starts with k=1 and if the objective does not improve, the neighborhood structure increases by on ($k \leftarrow k + 1$) and it resets k to one if $k > k_{max}$. On the other hand, the SA, unlike the VNS, accepts worse moves with a small probability in order to escape from being trapped in local optima. This probability is calculated by the Boltzmann function which uses Eq.(9).

$$Pr = e^{-\frac{C}{\alpha \times T}} \geq r \quad (9)$$

where C is change in the evaluation function (i.e. $C = F_{current} - F_{best}$),$\alpha$is a constant, and T is the current temperature. If Pr is greater than a random number, r, in interval [0, 1], it will accept a worse move. Moreover, in the proposed solution algorithm a long-term memory with a simple heuristic procedure is applied to store the visited moves and prevent them from being visited again.

3.1 Initial solution

With no doubt, the initial solution has an enormous impact on every solution algorithm. Our computational tests indicated that the optimal solution of each scenario (i.e. when we solve a deterministic CFLP model with data from that scenario s)in the most cases has a close gap with the optimal solution of the problem.As a result, we randomly take the solution of one of the scenarios as the initial solution.

3.2 Neighborhood generating procedure

The performance of the proposed algorithm significantly depends on the neighborhood structure and other related operators. Consequently, using an efficient scheme for neighborhood generation is crucial for a successful algorithm implementation. As we discussed earlier, the VNS randomly replaces k basic nodes with k non-basic nodes and in each iteration when the objective function does not improve, it increases the neighborhood structure by one until k become greater than Kmax or the objective function improves which resets k to one. We tested this procedure to solve our model but the performance of the algorithm was very weak. As majority of the improvement moves are visited when the algorithm uses the first neighborhood structure, we modify the scheme of changing the neighborhood structure. Accordingly, we increase the neighborhood structure if and only if the current move does not improve the objective function and satisfies the following equation.

$$\exp\left(-\frac{C}{T_0}\right) \geq random(0, 1) \quad (10)$$

Additionally, in each iteration we use the following cooling equation.

$$T_0 \leftarrow \alpha \times T_0 \quad (11)$$

3.3 *Local* search

In this article, we use an exact method to return the local optima of the generated solutions. In better words, we use the software developed by Ferris (2005)that makes an interface between MATLAB and GAMS, in which MATLAB passes the generated moves to GAMS and then using the CPLEX solver , GAMS determines their local optima while the location variables are fixed.

3.4 Stopping criteria

The algorithm stops whenever it meets one of the following criteria:

- ✓ *Elapsing the considered maximum CPU time*
- ✓ *Achieving the optimal solution*
- ✓ *The objective function does not improve after 3*|J| iterations.*

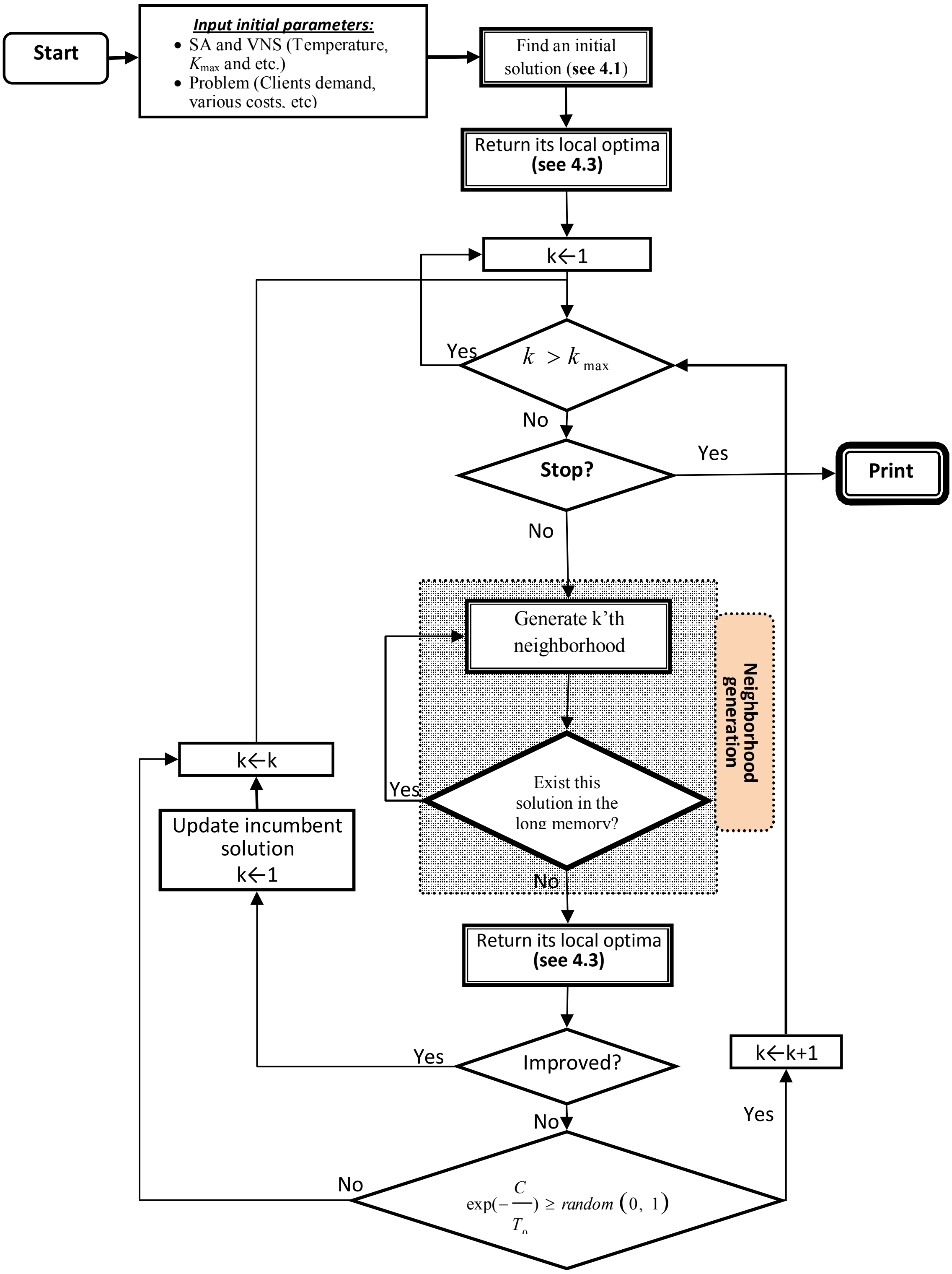

Fig. 1. The flow chart of the proposed algorithm

4. Numerical Experience

In this part, we present the result of our computational study and analyze the algorithm's efficiency in terms of CPU time requirement and optimality gap. The developed model was implemented on a range of test problems solved with the standard mathematical programming software GAMS 23.3.3, namely with the branch-and-bound algorithm of CPLEX 12.1, and the algorithm was coded in MATLAB 7.6 on a Personal DELL computer with a 2.22GHz processor and 3GB of RAM . Note that in all data sets each node serves as both a customer and a potential facility site (i.e., I =J).

4.1 Data generation

To test the efficiency of our algorithm, we solve a variety of test problems which include both standard and randomly generated test problems. The standard test problems are taken from the original authors' website, Snyder and Daskin (2006). That means for the standard test problems the data used for the uncertain parameters, i.e.f_i^s,D_i^s, and l_{ij}^s, are taken from he reference(Snyder and Daskin 2006). These test problems have 49 and 55 nodes with 9 and 5 scenarios, respectively.

In the generated test problems, the locations of the costumers are generated randomly and uniformly distributed over an100×100 area.In each data set, demands and fixed costs for scenario-1 are drawn uniformly from [0, 10000] and [4000, 8000], respectively. Additional scenarios are generated by multiplying scenario-1 data by a random number drawn uniformly from [0.5, 1.5]. Travel distances between the facilities and the customers for scenario-1 are considered equal to their Euclidean distances, and other scenarios are obtained by multiplying random numbers drawn uniformly from [0.5, 1.5] by the scenario-1 data. In order to test the algorithm's capabilities any further, we generated symmetrical travel distances in all scenarios whereas these values f are nonsymmetrical or the standard test problems.

For the sake of convenience, we assume that all scenarios have an equal occurrence probability, or $q_s = {}^1\!/_S$. In addition, facilities' capacity is computed by means of Eq.(12). In other words, the maximum demand over all scenarios is determined and then is divided by the maximum allowable number of facilities. Since the number of open facilities is equal to or lower than the V facilities, it should be possible that the model determines the number of optimal open facilities between 1 to V facilities. Thus, an arbitrary value(Ψ) is added to the adequate capacity for V open facilities.

$$C = \frac{max_{s\in S}\{\sum_{i\in I} D_i^s\}}{V} + \Psi \qquad (12)$$

4.2 Computational results

The developed model has not been considered formerly in the literature. Accordingly, we cannot compare our algorithm's efficiency to the other solution algorithms and, in turn, we compare it with the CPLEX solver which is the most powerful solver for mixed-integer models.

In the present paper, we have fixed the maximum CPU run time at 2000 seconds for the test problems with up to 50 nodes and for the other instances at 4000 seconds. Ten and fifteen percent of networks nodes, $V = (10\%\ or\ 15\%) \times |J|$, are considered as the maximum number of facilities. Finally, in two cases the proposed algorithm is compared with the CPLEX. Accordingly, the computational results of the discussed test problems for the minimax regret formulation (i.e. Eq.(1, 3, 4, 6, 7, and 8))are reported in Table 3 and those of the min-expected-cost objective function when p=∞ (i.e. Eq.(2, 3, 4, 5, 6, 7, and 8)) are presented in Table 4.

In these tables, the labeled columns under "Test Problems"with "TP", "No"., "Fac", and "Scen" represent the number of test problems, number of nodes, number of facilities, and number of scenarios, respectively. Moreover, the columns under "CPLEX" which are labeled "Lower bound","GAP%", and" CPU Time"symbolize the lower bound, the relative error between the final solution and the lower bound that is computed by means of Eq. (13) and the elapsed CPU time in seconds, respectively. Furthermore, the column under "Hybrid Algorithm" which is labeled with "Bcpu" reports the elapsed time when the best solution is obtained by the algorithm. In addition, the average CPU time requirements in seconds and the average gap are listed in the last rows.

$$\boldsymbol{Gap}\% = \frac{\boldsymbol{Obj.-LB}}{\boldsymbol{LB}} \times \boldsymbol{100} \qquad (13)$$

Note from Table 3 that for those test problems with zero gap(TP1 and TP2), on average the heuristic found the optimal solutions in less than 12.65 seconds whereas the CPLEX required 192.50 seconds.For other instances (TP3-7), on average the heuristic reported results that by using 1144.11 seconds have 31.28 % gap while the CPLEX reported solutions with 51.32 % gap by elapsing 3200 seconds. In other words, the VNS was able to reduce the average gap and the average elapsed time of the CPLEX more than 39.06 % and 64.25 % respectively.

Table 3
Performance comparison: under minimax regret formulation

Test Problems				*CPLEX*			*Hybrid Algorithm*	
TP	No.	Fac.	Scen.	Lower bound	GAP%	CPU Time (s)	GAP%	Bcpu(s)
TP1	10	1	10	0.1082	0.00	3.31	0.00	3.11
		2	10	0.8147	0.00	9.95	0.00	3.23
TP2	20	2	10	0.3867	0.00	94.21	0.00	7.94
		3	10	0.2311	0.00	662.54	0.00	36.30
TP3	40	4	10	0.1349	19.52	2000	16.98	35.08
		6	10	0.2836	63.11	2000	18.39	473.77
TP4	49	5	9	0.0232	4.34	2000	4.34	420.91
		8	9	0.0234	0.86	2000	0.86	275
TP5	55	5	5	0.0737	35.18	4000	13.40	1080.93
		9	5	0.0661	31.09	4000	25.47	605.95
TP6	60	6	10	0.1040	78.42	4000	40.87	562.36
		9	10	0.0910	89.17	4000	70.47	1538.33
TP7	80	8	10	0.0606	95.53	4000	68.97	2925.61
		12	10	0.0785	96.00	4000	53.02	3523.14
AV.					36.66	2340.72	22.34	820.83

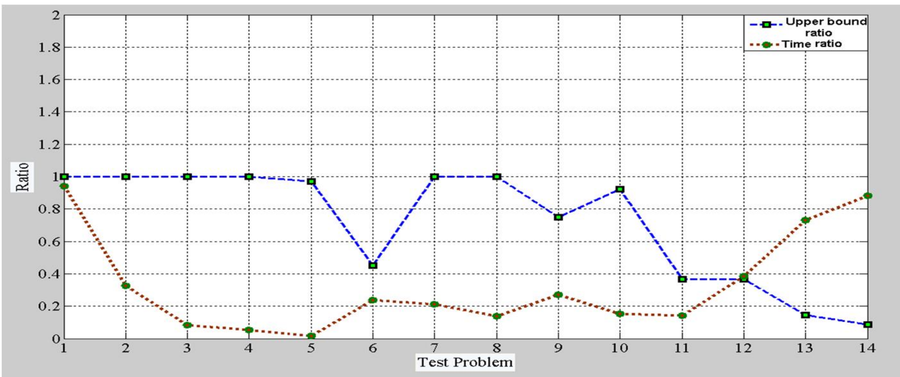

Fig. 2. Performance comparison of the proposed algorithm with CPLEX solver

It is also interesting to compare the obtained feasible solutions with each other in addition to the gap comparison. To do so, we divided the algorithm results by the CPLEX results. The obtained ratio shows a fraction of CPLEX's results which has been obtained by the algorithm. For example, 0.40 indicates that the algorithm has been able to improve the CPLEX's result by 60%. The computed results are depicted in Figure 2 with "upper bound ratio"and "Time ratio" legends. Figure 2 obviously shows that the algorithm has performed much better for all test problems.

We also tested the attractiveness of our algorithm in
The obtained .solving min-expected-cost objective functions
gaps and elapsed times are reported in

Our algorithm was able to achieve the optimal .Table solution for those instances that their optimality was proved For these instances on average the .2)-by the CPLEX(TP1 92 seconds of CPU time whereas the .algorithm used 7 -For other instances(TP3.76 seconds.CPLEX required 40 TP7), the average time and the average gap obtained by the 72, whereas these values for .44 and 3.algorithm were 1184 34respectively.Note from .the CPLEX were 3200 and 6

Table that for the TP7 with 5 facilities the CPLEX failed to achieve any initial feasible solution while the algorithm was able to find a solution that is close to the lower bound (i.e., 9.08%).On average, the proposed algorithm obtained 2.66%gap within849 seconds and the CPLEX was able to achieve solutions that have 4.39%gap within 2297.36 seconds.

Table 4
Algorithm performance for infinite robustness coefficient (p=∞)

Test Problems				*CPLEX*			*Hybrid Algorithm*	
TP	No.	Fac.	Scen.	Lower bound	GAP%	CPU Time (*s*)	GAP%	Bcpu(*s*)
TP1	10	1	10	1267514.65	0.00	3.45	0.00	13.21
		2	10	763966.02	0.00	4.43	0.00	4.08
TP2	20	2	10	2654621.79	0.00	97.31	0.00	2.18
		3	10	1993644.51	0.00	57.86	0.00	22.20
TP3	40	4	10	3234733.69	2.30	2000	1.36	1042.62
		6	10	2448367.03	12.99	2000	3.36	762.34
TP4	49	5	9	27791.27	0.11	2000	0.09	653.02
		8	9	42648.34	4.38	2000	4.38	111.21
TP5	55	5	5	14502.71	1.66	4000	1.22	116.79
		9	5	15079.76	2.98	4000	0.96	1037.91
TP6	60	6	10	3363407.22	8.17	4000	5.89	1077.65
		9	10	2496276.67	5.04	4000	5.04	762.54
TP7	80	8	10	4077224.66	"N/A"	4000	9.08	3115.61
		12	10	3035864.14	19.46	4000	5.82	3164.75
AV.					4.39	2297.36	2.66	849.00

4.3 Behavior comparison between the algorithm and the CPLEX

For the standard test problem with 49 nodes and 5 facilities (TP4), we investigated the behavior of both CPLEX and the proposed algorithm in improving their solutions' quality and their convergence speed over time. As a result, the CPU times when the solutions improved are recorded and the gap of these values with the best known lower bound (seeTable3) are calculated. These values are depicted in Figure 3.As this plot shows, the proposed initial solution has less optimality gap in comparison with the first feasible solution of the CPLEX. Moreover, after 1600 seconds the CPLEX improves its solution for the last time while the algorithm has found the same solution quality in 420 seconds.

4.4 Cost vs. Regret

One of the main purposes of the developed model is to design a more robust system.In other words, a large reduction in the maximum regret makes a little increase in the expected objective function (costs). As it was pointed out by Snyder and Daskin(Snyder and Daskin, 2006), a more robust design can be bought with small increases in the expected costs. Similar phenomena can be observed for the developed model in this study.In line with this subject, we first solved the model with the infinite robustness coefficient, and then we calculated the maximum relative regret over all scenarios. Subsequently, the p value the problem was re-solved.This process was continued until no feasible solution could be found. We performed this process on the TP2(V=2). We observed that with 15.71%reduction in the maximum regret the expected cost increases only by1.94 %. In addition, reducing p value from infinite to the minimum value for the TP4 did not cause any increase in the expected costs.

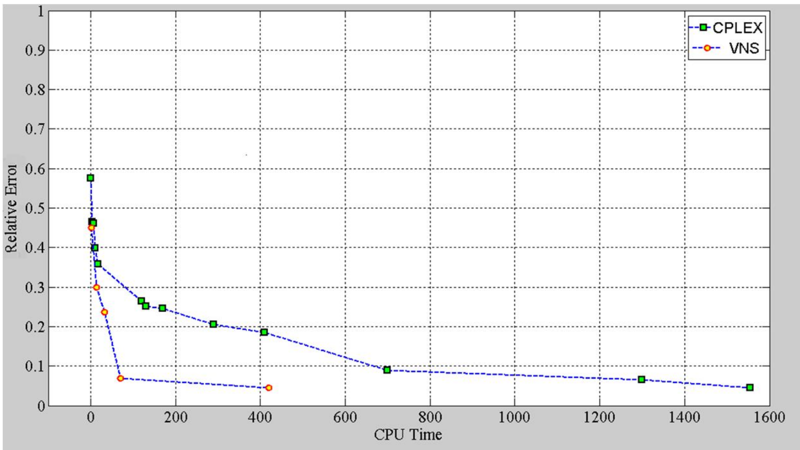

Fig.3.Convergence speed comparison between CPLEX and the Algorithm

5. Conclusion

In this paper an extension of the capacitated facility location problem under uncertain environment was studied. We discussed both stochastic and robust formulation of the problem and used the stochastic p-robust approach to face the uncertainty in the parameters. The objective function of the developed model minimized the total expected costs of operating and transportation while the relative regret in each scenario was restricted. In order to solve the model, we also proposed a fix-and-optimize hybrid algorithm. This algorithm uses several meta-heuristic algorithms to generate neighborhoods and uses an exact method to find the local optima. Additionally, to test the algorithm’s performance, we applied numerical experiments on a wide range of both standard and randomly generated test problems. The computational results demonstrated that the proposed algorithm, besides being simple, was very efficient and attractive in terms of CPU time requirement and solution quality in comparison with the CPLEX solver.Further Studies may take into account other real assumptions such as disruption, congestion, queening, inventory, and so forth. Furthermore, developing other solution algorithms and an efficient local search algorithm to solve larger instances seem to be appropriate future research avenues.

6. Acknowledgments

We wish to thank two anonymous reviewers for their productive comments on this paper which greatly enhanced both the insights and presentation of the paper.

7. References

[1] Barahona, F. and Chudak, F. A., (2005). "Near-optimal solutions to large-scale facility location problems." Discrete Optim 2: 35–50.

[2] Beasley, J. E., (1993). "Lagrangian heuristics for location problems." European Journal of Operational Research 65: 383-399.

[3] Brandeau, M. L., and Chiu, S. S., (1989). " An Overview of Representative Problems in Location Research." Management Science 35(6): 645-674.

[4] Brimberg, J. and Mladenovi´c, N., (1996). " A variable neighborhood algorithm for solving the continuous location-allocation problem." Studies in Locational Analysis 10: 1-12.

[5] Cornuejols, G., Sridharan, R., (1991). "A comparison of heuristics and relaxations for the capacitated plant location problem." European Journal of Operational Research 50: 280-297.

[6] Ferris, M. C., (2005). "MATLAB and GAMS: Interfacing Optimization and Visualization Software, University of Wisconsin, http://www.cs.wisc.edu/math-prog/matlab.html."

[7] Fleszar, K. and Hindi, K. S., (2008). "An effective VNS for the capacitated p-median problem." European Journal of Operational Research 191: 612-622.

[8] Ghaderi, A., Jabalameli, M. S., (2011). "An efficient hybrid particle swarm optimization algorithm for solving the uncapacitated continuous location-allocation problem." Netw Spat Econ Accepted for publication, doi:10.1007/s11067-011-9162-y.

[9] Ghiani, G., Guerriero, F., et al., (2002). "The capacitated plant location problem with multiple facilities in the same site." Computers & Operations Research 29(13): 1903–1912.

[10] Hansen, P., Mladenovi´c, N., et al., (2010). "Variable neighbourhood search: methods and applications." Annals of Operations Research 175: 367–407.
[11] Hindi, K. S. and Pienkosz, K., (1999). "Efficient solution of large scale, single source, capacitated plant location problem." Journal of the Operational Research Society 50: 268–274.
[12] Korupolu, M., Plaxton, C., et al., (2000). "Analysis of a local search heuristic for facility location problems." Journal of Algorithms 37: 146-188.
[13] Küükdeniz, T., Baray, A., et al. (2012). "Integrated use of fuzzy c-means and convex programming for capacitated multi-facility location problem." Expert Systems with Applications 39(4): 4306-4314
[14] Leitner, M. and Raidl, G. R., (2012). "Variable Neighborhood and Greedy Randomized Adaptive Search for Capacitated Connected Facility Location " Lecture Notes in Computer Science DOI: 10.1007/978-3-642-27549-4_38.
[15] Manzour-al-Ajdad, S. M. H., Torabi, S. A., et al. (2012). "Single-Source Capacitated Multi-Facility Weber Problem—An iterative two phase heuristic algorithm." Computers & Operations Research 39: 1465–1476.
[16] Marianov, V. and Serra, D., (2002). Location problems in the public sector. Facility Location: Applications and Theory. Z. Drezner and H. W. Hamacher. Berlin, Springer.
[17] Melkote, S. and Daskin, M. S., (2001). "An integrated model of facility location and transportation network design." Transportation Research Part A 35: 515-538.
[18] Murali, P., Ordóñez, F., et al. (2012). "Facility location under demand uncertainty: Response to a large-scale bio-terror attack." Socio-Economic Planning Sciences 46: 78-87.
[19] Nauss, R. M. (1978). "An improved algorithm for the capacitated facility location problem." Journal of the Operational Research Society 29: 1195-1201.
[20] Nikoofal, M. E. and sadjadi, S. J., (2010). "A robust optimization model for p-median problem with uncertain edge lengths." International Journal of Advanced Manufacturing Technology: 1-7.
[21] Owen, S. H. and Daskin, M. S., (1998). "Strategic facility location: A review." European Journal of Operational Research 111: 423-447.
[22] Pahlavani, A. and Saidi-Mehrabad, M., (2011). "A Fuzzy Multi-objective Programming for Optimization of Hierarchical Service Centre Locations " Journal of Uncertain Systems 5(3): 202-226.
[23] Pasandideh, S. H. R. and Chambari, A., (2010). "A New Model for Location-Allocation Problem within Queuing Framework " Journal of Industrial Engineering 6: 53-61.
[24] Rahmaniani, R., Ghaderi, A., et al. "Stochastic p-robust uncapacitated multiple allocation p-hub location problem." International Journal of Industrial and Systems Engineering Articel In Press.
[25] Sa, G., (1969). "Branch and bound and approximate solutions to the capacitated plant location problem." Operations Research 17: 1005-1016.
[26] Snyder, L. V., (2006). "Facility location under uncertainty: a review." IIE Transactions 38: 537-554.
[27] Snyder, l. V. and Daskin, M. S., (2006). "Stochastic p-robust location problems." IIE Transactions 38: 971-985.
[28] Sun, M. (2011). "A tabu search heuristic procedure for the capacitated facility location problem." Journal of Heuristics Article in press: 1-28.
[29] Yin, P. and Mu, L., (2012). "Modular capacitated maximal covering location problem for the optimal siting of emergency vehicles." Applied Geography 34: 247–254.