Journal of Uncertain Systems
Vol.7, No.1, pp.22-35, 2013
Online at: www.jus.org.uk

Robust Capacitated Facility Location Problem: Optimization Model and Solution Algorithms

Ragheb Rahmaniani*, Mohammad Saidi-Mehrabad, Hojjat Ashouri
Department of Industrial Engineering, Iran University of Science and Technology, P.C. 1684613114, Narmak, Tehran, Iran

Received 6 July 2011; Revised 13 November 2012

Abstract

In this article, we propose an extension of the capacitated facility location problem under uncertainty, where uncertainty may appear in the model's key parameters such as demands and costs. In this model, it is assumed that facilities have hard constraint on the amount of demand they can serve and, as a result, some customers may not be fully satisfied. Unfortunately, traditional models ignore this situation and if facilities do not serve all demands, the model becomes infeasible. Accordingly, we develop the mathematical formulation in order to allow partial satisfaction by introducing penalty costs for unsatisfied demands. In general, this model optimizes location for predefined number of capacitated facilities in such a way that minimizes total expected costs of transportation, construction, and penalty costs of uncovered demands, while relative regret in each scenario must be no greater than a positive number ($p \geq 0$). The developed model is NP-hard and very challenging to solve. Therefore, an efficient heuristic solution algorithm based on the variable neighborhood search is developed to solve the problem. The algorithm's efficiency is compared with the simulated annealing algorithm and CPLEX solver by solving variety of test problems.Computational experiments show that the proposed algorithm is more effective and efficient in terms of CPU time and solutions quality.
© 2013 World Academic Press, UK. All rights reserved.

Keywords: capacitated facility location problem, uncertainty, robust optimization, partial satisfaction, variable neighborhood search, simulated annealing

1 Introduction

Location problems are critical managerial decision with a large body of literature and numerous applications in the real-world application [5, 9, 10, 24, 26]. In fact, facility location decisions are long-term strategical decisions and almost impossible to reverse. These strategic decisions have a great impact on the network's flows and customers' satisfaction. Essentially, these problems deal with locating a number of facilities to supply a set of customers at the minimum cost with respect to several constraints and involve various objectives which naturally are in conflict [27]. For example, in some location problems there might be hard constraints on amount of available budget, number of facilities, facilities' capacity, coverage distance, and so forth, while the objective function maximizes total covered demand and minimizes total construction and transportation costs.

In line with this subject, numerous papers have been published in which the Capacitated Facility Location Problem (CFLP) is one of the most important and basic models. This problem has been intensively considered in the literature, since it includes more realistic assumptions in compare with, for example, uncapacitated location problems. In the CFLP, a number of capacitated facilities, P, are to be located among J possible sites in order to satisfy demands of I customers by minimizing total costs of transportation and fixed charges of establishing facilities. The CFLP is NP-hard problem which is generalized from the simple plant location problem. The CFLP has been effectively implemented to solve real-world applications such as plants location, power stations location, warehouses location, to just name a few. Though this problem and its variations have been intensively studied in the literature, this problem has been mainly investigated with deterministic data and with this assumption that facilities' capacity can meet all customers' demand.

* Corresponding author. Email: ragheb.rahmaniani@gmail.com (R. Rahmaniani).

However, in the reality, these assumptions are rarely satisfied. Utilized data in managerial decisions are encountered within complete factors such as noisy and erroneous defects. Meanwhile, in many cases due to lack of historical information, it is almost impossible to obtain a reliable and certain estimation for the model's key parameters. On the other hand, location decisions are long-term strategic decisions and very sensitive to the changes in demands and other parameters. Therefore, ignoring uncertainty in parameters yields in ineffective decisions. As a result, optimization under uncertainty has received increasing attention in the location theory during the last few decades [2, 13, 32, 30, 35] and, various approaches are proposed to face and resolve the uncertainty. These approaches can be generally categorized into robust and stochastic optimization approaches.

In the robust approach, the uncertain parameters are estimated by discrete or interval data and the objective function typically minimizes minmax cost or minmax regret. The regret in each scenario is defined as distance between the current cost of that scenario and its optimal cost. Discrete scenarios are used when probability distribution for uncertain parameters is not known. Robust solutions are naturally too conservative decisions and impose unnecessary long-term costs. On the other hand, stochastic programming models partition decisions in two-stages in which objective function minimizes the sum costs of the first-stage and the expected costs of the second-stage. In general, the typical stochastic programming may yield inexpensive solutions in the long run but perform poorly under certain realizations of the random data. Besides, the robust problems due to the minmax structure are much harder to solve. Based on these weaknesses, Snyder and Daskin [32] introduced the *stochasticp-robust* optimization approach which comes through combining main concepts of these approaches. One of the main goals of this approach is to design a more robust system with a little increase in the expected costs. In this study, this approach is applied to optimize the CFLP under uncertainty. Considering uncertainty increases reliability of solutions.

Moreover, traditional CFLP models become infeasible if total opened capacity is lower than sum of customers' demand. Whereas, in the reality, it is likely that facilities due to their limited capacity leave some of these demands unsatisfied. For instance, power stations have hard constraint on the amount of generated electricity and they cannot provide unbounded amount of electricity to the customers. Moreover, due to the huge investment cost of establishing power stations, managers cannot construct more than a limited number of power stations. Hence, with a limited number of capacitated power stations, logically, we will not be able to supply all customers' increasing demand. This situation can be observed in third-world and deprived area by far. Unfortunately, traditional models of CFLP cannot find even a feasible solution for such problems. Therefore, we also develop the mathematical formulation of CFLP with this assumption that our facilities may not be able to supply all demands and any proportion of customers' demand may remain unsatisfied. On the other hand, losing any proportion of demand imposes a great deal of cost to the company, since they lose the market share. As a result, the objective function minimizes penalty costs of unsatisfied demands, as well.

Generally speaking, the main goal of this study is to develop the mathematical formulation of CFLP under uncertainty by taking these facts into account that each customer may not be fully supplied and their demands are uncertain and characterized by a given set of scenarios. Moreover, we associate a penalty cost to any proportion of unsatisfied demand. Therefore, the objective function minimizes total costs of establishing new facilities, expected costs of transportation, and expected penalty costs of unsatisfied demands. The developed model will be a two-stage model in which the first-stage optimizes facilities' location and the second-stage optimizes customers' assignment. Each customer may be supplied by several facilities (i.e. multiple assignment property) and any proportion of its demand may be satisfied (i.e. say 60% of its total demand). This formulization will be referred to us as the Stochastic p-robust Capacitated Facility Location Problem, or *p*-SCFLP as an abbreviation. In addition, since the developed model formulation is hard and cannot be effectively solved using ordinary optimization methods, we propose solution algorithms based on Variable Neighborhood Search (VNS) and Simulated Annealing (SA) to solve the developed model. We test the efficiency of these algorithms on a variety test instances and compare their result with the CPLEX solver in terms of optimality gap and CPU time requirement.

The reminder of this paper is organized as follow: in the next Section the related literature is reviewed. In Section 3 the mathematical formulation of the *p*-SCFLP is presented. Subsequently, solution approaches based on the VNS and SA are outlined in Section 4. Likewise, our numerical experiments are summarized in the Section 5 and finally, our conclusions and future research directions are discussed in the Section 6.

2 Related Literature

The CFLP is a classical facility location problem with large body of the literature. The CFLP has been extensively investigate don the both side of model formulation and solving algorithms. Various algorithms that have been implemented to solve the Uncapacitated Facility Location Problems (UFLP) are usually generalized to solve the

CFLP. For example, Kuehn and Hamburger [22] proposed an algorithm to solve the UFLP and then Jacobsen [17] extended this algorithm for the CFLP. However, in this section, we are going to address the most relevant papers to the considered problem and the interested readers are referred to [1, 26, 33] and references therein.

Lagrangian Relaxation (LR) has been widely considered as an efficient solution algorithm to solve the CFLP. Context presented by Cornuejols et al. [7] provid an excellent theoretical analysis of all possible Lagrangian relaxations and the linear programming relaxation for the CFLP. A Lagrangian Heuristic (LH) framework is presented by the Beasley [4] to solve different facility location problems. In the proposed method for the CFLP, allocation constraints and capacity constraints are incorporated into objective function by using Lagrangian multipliers. In addition, LH method for the both UFLP and the CFLP is also proposed by other authors [3, 23]. Barahona and Chudak [3] initially provided the linear programming relaxation of the CFLP and then proposed the Lagrangian relaxation to solve the linear problem. Heuristics and meta-heuristics algorithms are also applied to solve the CFLP in a wide range. Sun [34] applied the Tabu Search (TS) to solve the CFLP and compared it with the Lagrangian and the surrogate/Lagrangian heuristic methods. He used long term memory based on primogenitary linked quad tree to store visited solutions and prohibits them from being visited again. Cortinhal and Captivo [8] proposed upper and lower bounds for the single source capacitated location problem. In that article, Lagrangian relaxation is used to obtain lower bounds for this problem and upper bounds are obtained by Lagrangian heuristics followed by search methods (e.g. TS). A simple local search heuristic for the capacitated facility location problem is presented by Korupolu et al. [20]. Hindi and Pienkosz [15] presented a heuristic that combines Lagrangian relaxation with restricted neighborhood search for the capacitated single source location problem. In this paper, a heuristic procedure with three phases is used to find feasible solutions. An efficient VNS heuristic for the capacitated P-median problem is investigated by Fleszar and Hindi [12]. They also proposed several intermediate heuristic search algorithms. These algorithms were able to find very good solutions within much shorter computation time than the full VNS algorithm .Similarly, exact methods have been investigated to solve the CFLP. Basically, the proposed exact methods are based on branch and bound and set partitioning technique with differences in strategies to increase the lower bound and type of relaxation. For instance, Sa [28] relaxed the CFLP to obtain transportation problem but Nauss [25] relaxed capacity constraints and added set of surrogate constraints to obtain tighter bounds. Nevertheless, interested readers are referred to the context by Sridharan [33] to get comprehensive review on the solutions algorithm for the CFLP. However, the published works related to CFLP have mostly concentrated on solution algorithms, and extensions of mathematical formulation are usually introduced as newer works such as capacitated hub location, capacitated maximal covering, capacitated facility location and network design, and so forth.

Facility location problems are intensively studied under uncertainty [2, 13, 16, 32, 30, 35]. For instance, Ghezavati et al. [13] proposed a robust approach to location-allocation problem under uncertainty. Snyder and Daskin [32] proposed a new approach for optimizing facility location problems under uncertainty. They combined the minimum-expected-cost and *p*-robustness measures together in order to introduce the *stochastic p-robust* optimization model. This approach is implemented on the classical models of the UFLP and P-median problem in order to resolve the uncertainty in demands and distances. They intended to find solution that has minimum-expected-cost while the obtained solution is *p*-robust; i.e., cost under each scenario for each feasible solution must be no greater than $100(1+p)\%$ of the optimal cost of that scenario, where p is a non-negative value known as robustness coefficient. They solved the proposed models by using Lagrangian decomposition and reducing them to the multiple-choice knapsack problem. Additionally, they discussed a mechanism for detecting infeasibility. The interest readers are referred to the [5, 21, 26, 29, 31] to get more information about location and location under uncertainty.

3 Problem Description and Formulation

Consider a production-distribution network that has $|I|$ major centers of customers and $|J|$ potential facility sites. A company wants to locate $|P|$ facilities in this network in order to service these customers. These facilities cannot supply unbounded amount of demand for customers, and consequently, some customers may not be fully satisfied or even may remain completely or partially unsatisfied. On the other hand, inability in satisfying customers' demand incurs huge costs for the company in consequence of losing the market share. As a result, we define penalty cost for unsatisfied demands as a function of unmet-demand. Therefore, if $X_{i,j}^{s}$ denotes fraction of demand of customer i which is satisfied by facility at node j under scenario s, we will have $1-\sum_{j\in J} X_{i,j}^{s}$ percentage of its demand unsatisfied in scenario s. Where, $\sum_{j\in J} X_{i,j}^{s}$ indicates total fraction of demand of customer i which is satisfied by facilities through the network. Accordingly, we define a variable in order to capture the proportion of demand of customer i under scenario *s* which is unsupplied, i.e., Z_i^s. By keeping this in mind that sum of total fraction of satisfied and unsatisfied demand for each customer in each scenarios should be equal to one or, one hundred percent of its demand, we have equation (1).

$$\sum_{j\in J} X_{i,j}^s + Z_i^s = 1 \qquad \forall i \in I,\ \forall s \in S. \tag{1}$$

Logically, each customer may be assigned to several facilities and may be partially supplied (say 80 percentage of its total demand). This is more close to the reality because in the classical models if facilities could not supply all demand, the model become infeasible, while there is still a suboptimal solution to the problem. To simplify the presentation of the mathematical formulation in this paper, the following notations are defined.The index sets and model parameters are described in Table 1.

Table 1: Parameters and index sets

Symbols	***Indexed by***	***Description***
J	$j \in \{1,2,...,\lvert J\rvert\}$	Set of potential facility site;
I	$i \in \{1,2,...,\lvert I\rvert\}$	Set of costumers;
S	$s \in \{1,2,...,\lvert S\rvert\}$	Set of scenarios;
Parameters	***Description***	
C	Facilities' capacity	
ω_s	Probability of that scenario *s* occur	
ρ	Desired robustness coefficient	
W_i^s	Demand at node *i* under scenario *s*	
f_j	The fixed charge of opening a facility on node *j*	
P	Number of facilities to be open	
d_{ij}^s	Travel distance between nodes *i* and *j* under scenario *s*	
Q_s^*	Optimal cost of CFLP under data from scenario *s*	
a	A constant number	

where Q_s^* is the optimal objective value of the deterministic CFLP problem with partial satisfaction (see equations (15-21)) under data from scenario *s* that can be computed by CPLEX or any exact method. The decisions of the stochastic *p*-robust capacitated facility location include decisions about locating facilities and assigning costumers to these facilities in such a way that total expected costs being minimized. These decisions are made in two stages and are defined as follows.

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

$X_{i,j}^s \geq 0$ fraction of demand of node *i* under scenario *s* that is satisfied by facility located at node j

$Z_i^s \geq 0$ fraction of demand of node *i* under scenario *s* that is notsatisfied.

Note that, here the location decision variables (y_j), unlike the assignment variables ($X_{i,j}^s$ and Z_i^s), are independent on the *s* index in order to reflect the two-stage nature of the problem. Finally, we assume that uncertainty is associated with demands and distances. Using this notation and assumptions, the proposed two-stage mixed-integer model for the problem in hand is formulated as follows.

$$Min\ \rho - SCFLP: \quad \sum_{j\in J} f_j\, y_j + \sum_{s\in S}\omega_s\left[\sum_{i\in I}\sum_{j\in J} W_i^s\, d_{ij}^s\, X_{i,j}^s + \sum_{i\in I} a\, W_i^s\, Z_i^s\right] \tag{2}$$

Subject to:

$$\sum_{j\in J} X_{i,j}^s + Z_i^s = 1 \qquad \forall i\in I,\ \forall s\in S \tag{3}$$

$$\sum_{i\in I} W_i^s\, X_{i,j}^s \le C\, y_j \qquad \forall j\in J,\ \forall s\in S \tag{4}$$

$$\sum_{j\in J} f_j^s\, y_j + \sum_{i\in I}\sum_{j\in J} W_i^s\, d_{ij}^s\, X_{i,j}^s + \sum_{i\in I} a\, W_i^s\, Z_i^s \le (1+\rho)Q_s^* \qquad \forall s\in S \tag{5}$$

$$\sum_{j\in J} y_j = P \tag{6}$$

$$Z_i^s \ge 0 \qquad \forall i\in I,\ \forall s\in S \tag{7}$$

$$X_{i,j}^s \ge 0 \qquad \forall i\in I,\ \forall j\in J, \forall s\in S \tag{8}$$

$$y_j \in \{0,1\} \qquad \forall j\in J. \tag{9}$$

In the proposed model, equation (2) represents the objective function. First term in this equation minimizes the fixed cost of opening facilities. Second term minimizes total expected transportation costs. Finally, the third term minimizes total expected penalty costs of unsatisfied demand. Penalty cost for each customer i is calculated as a function of lost demand ($W_i^s * Z_i^s$) multiple to a constant number. Here, we have assumed that there is linear relation between lost demand and costs imposed to the company, while, in the reality, it is likely that penalty cost be a power function of lost demand as equation (10),

$$a * \left(W_i^s * Z_i^s\right)^b. \tag{10}$$

However, for the sake of simplicity in solving the model, we have set b=1. Otherwise, the model would become a nonlinear mixed-integer formulation which is much more challenging to solve. Moreover, in the objective function, ω_s assigns weight to each scenario which indicates importance of that scenario in decision making, where $\sum_{s\in S}\omega_s = 1$. Constraint (3) guarantees that sum of total satisfied and unsatisfied demand for customer i in each scenario do not exceed than its total demand.

Equation (4) represents the capacity constraint in which we ensure that the total allocated demand to the facility at node j in scenarios (i.e. $\sum_{i\in I} W_i^s * X_{i,j}^s$) does not exceed its capacity. Additionally, this constrain indicates that customer should be only assigned to the facilities, i.e., $X_{i,j}^s \le y_j$. Equation (5) enforces the p-robustness criterion. Accordingly, the objective function of each scenario s should not be greater than $(1+\rho)$% of optimal cost of that scenario. Robustness coefficient can be set differently in each scenario and denote by ρ_s, in order to account for the fact that different scenarios may have different importance levels in our decision. However, for the sake of simplicity, we consider the same ρ value for each scenario. Additionally, this constraint indicates that we are willing to make additional investment in the infrastructure in order protect against future possible disruptions. Note that, this model would be equal to the classical capacitated P-median problem if $\rho = \infty$ (robustness constraint will be inactive), $|S| = 1$, a=0, and constraint (3) is replaced by the following equation

$$\sum_{j\in J} X_{i,j}^s = 1 \qquad \forall i\in I,\ \forall s\in S. \tag{11}$$

The minmax regret formulation which is a typical objective function of robust models is generally used where the scenarios' probability is not known. This objective function minimizes maximum regret over all possible scenarios. We can easily change the proposed model in order to obtain this formulation. To do so, we simply replace the objective function with the robustness coefficient ρ which yields in the following formulation. We call this formulation Robust Capacitated Facility Location Problem or, RCFLP.

$$Min\ RCFLP: \quad \rho \tag{12}$$

Subject to:

$$\rho \geq 0 \tag{13}$$

$$(3), (4), (5), (6), (7), (8), (9). \tag{14}$$

Since locating large number of facilities is not desirable, equation (6) limits number of facilities that can be opened. Finally, equations (7 and 9) declare type of assigning variables and location variables respectively.

The deterministic form of the proposed model formulation which would be used to determine optimal cost of each scenario is as follow.

$$Min\ DCFLP: \quad \sum_{j \in J} f_j\ y_j + \sum_{i \in I} \sum_{j \in J} W_i\ d_{ij}\ X_{i,j} + \sum_{i \in I} a\ W_i\ Z_i \tag{15}$$

Subject to:

$$\sum_{j \in J} X_{i,j} + Z_i = 1 \qquad \forall i \in I, \tag{16}$$

$$\sum_{i \in I} W_i\ X_{i,j} \leq C\, y_j \qquad \forall j \in J, \tag{17}$$

$$\sum_{j \in J} y_j = P \tag{18}$$

$$Z_i \geq 0 \qquad \forall i \in I, \tag{19}$$

$$X_{i,j} \geq 0 \qquad \forall i \in I, \forall j \in J, \tag{20}$$

$$y_j \in \{0,1\} \qquad \forall j \in J. \tag{21}$$

4 Solution Approach

For realistically sized instances it is very challenging to solve most of the *p*-SCFLP problems to optimality within a reasonable computer CPU time and memory by the well-known optimization solvers such as GAMS/CPLEX or Lingo. Therefore, in this part, two heuristic algorithms based on the VNS and SA are developed to solve the *p*-SCFLP. Variable neighborhood search is a simple and powerful solution framework to find near to optimal solutions to large scale and complex problems in a reasonable computer time. It was firstly proposed by Brimberg and Mladenović [6] to solve continues location allocation problem. Comprehensive review on the method and applications for this meta-heuristic is prepared by Hansen and Mladenović [14]. Likewise, SA that mimics cooling process of materials is another efficient solution algorithm to solve highly complex problems [36].

Broadly speaking, the *p*-SCFLP involves two main decision: (I) decision regarding facilities' location, (i.e., y_j) and, (II) decision about assigning customers to the located facilities (i.e. $X_{i,j}^s$ and Z_i^s). In the proposed model, if facilities' location was given, an optimal solution for the allocation sub-problem is easy to obtain. In other words, the problem can be easily solved under any given vector for the y_j. Therefore, the proposed algorithms in each iteration determine facilities' location, y_j, and then uses a local search to determine the local optima for the allocation sub-problem. This local search is based on an exact optimization algorithm that uses Branch & Bound and cutting plane methods in its framework. Additionally, since testing a move several times is not computationally efficient, in these algorithms, a long-term memory is applied to store visited moves and prevent them from being visited again.

In order to show a move vector, a $|J|$-digit binary vector with equal length to the total potential facilities site is used. The j^{th} cell on the chromosome vector indicates that at node *j* facility is located (1) or not (0) in the current solution. For example, Fig. 1 represents the move vector of a network that includes 10 potential facilities site and nodes 1 and 7 are selected in current solution to be open and other node to be close.

1	0	0	0	0	0	1	0	0	0
1	2	3	4	5	6	7	8	9	10

Figure 1: A chromosome for solution representation

Moreover, the GAP value in percentage (relative distance between solutions and best possible solution) is calculated by means of equation (22). In this equation, F_{final} indicates final solution obtained with a heuristic method, and F_{best} denotes the lower bound obtained by CPLEX.

$$\% GAP = \frac{F_{final} - F_{best}}{F_{final}} \times 100. \tag{22}$$

At the subsequent subsections the proposed algorithms are discussed in details and finally, the flowchart of proposed VNS and outline of SA are presented.

4.1 Construction of an Initial Solution

With no doubt, initial solution has a great impact on every solution algorithm. More importantly, the VNS algorithm uses the best recorded solution as the incumbent solution to generate next moves. Therefore, a close initial solution to the optimal solution has an enormous impact on a successful VNS implementation. Our computational tests indicated that optimal solution of each scenario has a close gap with the optimal solution of the proposed model. As a result, the initial solution is obtained from selecting the optimal solution of that scenario that has greater probability of occurrence.

4.2 Local Search

Local search is a method to return the local optima for the generated solutions. In this study software that makes an interface between MATLAB and GAMS is used to obtain the local optimal from CPLEX. This software was developed by Ferris [11]. In better words, algorithms are coded in MATLAB software and it generates the moves (see Figs. 2 and 3) and pass them to GAMS software and then the CPLEX solver will solve the problem while the location variables (i.e. y_j) are fixed. This method is simple and rather efficient.

4.3 Main Procedure of VNS Algorithm

The main procedure of the algorithm is illustrated in Fig. 2. The algorithm starts with an initial solution for the location of facilities and then determines its local optima by CPLEX solver.In the improvement phase, several parameters such as, pre-selected neighborhood structure (i.e. K_{max}) and set of solutions in the generated k^{th} neighborhood (i.e. $N_k(x)$) have extreme impact on the algorithm's performance and need to be carefully tenured. As a result, we examined several strategies and the best combination is discussed here.

For a given neighborhood structure, k, VNS randomly selects k basic nodes from the incumbent solution and replaces them with k non-basic nodes in order to generate a new solution. Our computational tests indicated that most of improvement moves are achieved when k is fixed at one. Consequently, we change the neighborhood structure after a predefined number of iterations without improvement (U_k). In the classical VNS, U_k is set to one; that means, after any non-improvement iteration neighborhood structure increases by one. Note that, this algorithm sets k to 1 if the objective function improved or $k > k_{\max}$.

Algorithm takes an initial solution as the incumbent solution, denoted by q_{best} , and finds the local optima for this solution (i.e. F_{best}) and whenever a move could improve the objective function they will be updated. Accordingly, it generates k'^{th} neighborhood from the best recorded move, denoted by q , and if this solution has not been tested previously, it will be checked to see if it improves the objective function. After applying the local search for this solution, denoted by $F_{current}$, following steps are applied in order to decide whether we can change the neighborhood structure or not, and the algorithm will be repeated until one of the stopping criteria perform.

If $F_{current} < F_{best}$ **then let** $\{q_{best} = q, k=1, \text{and UI}=0\}$
Else if $UI <= U_k$, **then let** $\{\text{UI=UI+1}, k = k, \text{and } q_{best} = q_{best}\}$
Otherwise let $\{k = k+1, \text{UI}=0, \text{and } q_{best} = q_{best}\}$

where, UI is a counter for iterations and, whenever $k > k_{max}$ it will be set to zero. Finally, the algorithm stops after elapsing the considered CPU time, having 3*|J| iteration without improvement, or achieving the optimal solution if it was known beforehand.

4.4 Main Procedure of SA Algorithm

Simulated Annealing (SA) is a probabilistic meta-heuristic that has widely been used to find reasonable solutions in a limited amount of time (see, [18, 19]). SA, unlike VNS, accepts worse moves with a small probability in order to escape from trapping in local optima. This probability is calculated by the Boltzmann function which uses equation (23).

$$\Pr = e^{-\frac{\Im}{\alpha \times T}} \geq r \tag{23}$$

where, C is change in the evaluation function (i.e. $\Im = F_{current} - F_{best}$), α is a constant, and T is current temperature. If Pr was greater than a random number, r, in interval [0, 1], it accepts worse move.

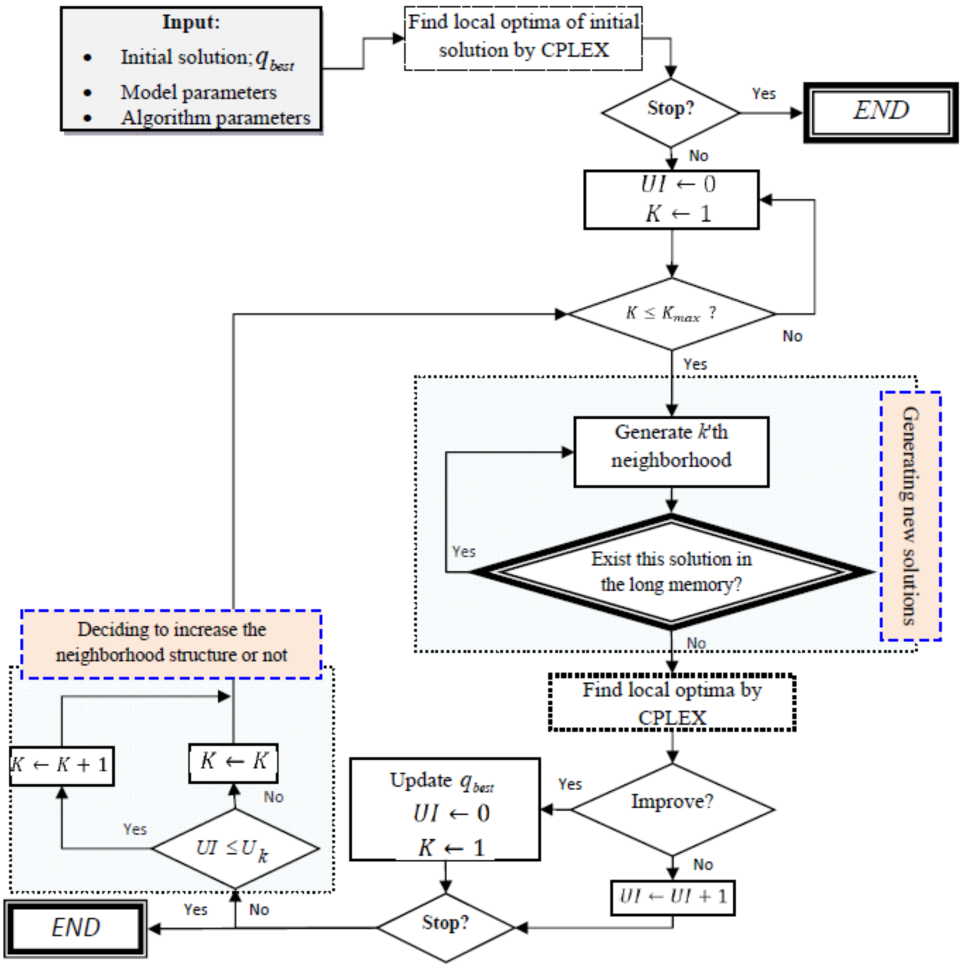

Figure 2: Main flow chart of the VNS algorithm

The pseudo-code of proposed SA algorithm to solve the model is outlined at Fig. 3. To begin with, an initial solution is obtained, then, the algorithm finds its local optima, denoted by F_{best}. At the step (2), the algorithm generates a new solution from X by randomly selecting one basic node and replacing it with one non-basic node. Accordingly, if the new solution improved the objective function both F_{best} and X will be updated. Moreover, if the objective function did not improve, we can still update X and accept worse move with a small probability. Finally, at each iteration temperature (i.e. T) is updated by using the cooling rate parameter α . This parameter is usually between 0.8 and 0.99. In this way, as time elapses the probability of accepting worse moves reduce. Moreover, we record the best achieved move and after a predefined number of iterations (say 100), the algorithms restarts from that solution if objective function did not improved. Finally, the algorithm is terminated in four cases: (1) the temperature to be less than TF, (2) the objective function is not improved during 3*|J| iterations, (3) the relative gap with the best possible solution become approximately zero and (4) maximum considered CPU run time elapses.

Initialize input parameters {model, α, T_0, *TF , stopping criteria}*
Step (1): *Calculate an initial solution: X*
Find a local optima for this solution X: F_{best}

Repeat following step until one of the stopping criteria perform
Step (2): *Generate a new solution Y from X based on k=1*
Step (3): *Apply the local search to find the local optima of Y:* $F_{current}$
Step (4): *Calculate change in the objective function:* $\Im = F_{current} - F_{best}$
If $\Im \le 0$ *then let {* $F_{best} \leftarrow F_{Current}$ *and* $X \leftarrow Y$ *}*
Otherwise {r = random(0, 1)}
If $\exp(-\frac{\Im}{T_0}) \ge r$ *then let {* $X \leftarrow Y$ *}*
Step (5): $T_0 \leftarrow \alpha \times T_0$

Figure 3: Outline of the proposed SA algorithm

5 Numerical Experience

In this part, experimental results are presented. The proposed problem was implemented on a range of test problems and were solved with standard mathematical programming software GAMS 23.3.3, namely with the branch-and-bound algorithm of CPLEX 12.1, and algorithms were coded in MATLAB 7.6 and run on a Core 2 Dual @2.22GHz DELL Notebook with 2GB RAM. In all data sets, each node serves as both a customer and a potential facility site (i.e., |I|=|J|).

5.1 Data Generation

To test the algorithms' performance, we applied numerical experiments on both randomly generated and standard test problems. The standard test problems were obtained from that is published by Snyder and Daskin [32]. These test problems have 49, 55, 88, and 150 nodes with respectively 9, 5, 9, and 9scenarios. We have summarized computational results for these instances in Tables 3 and 5. In the generated test problems, the locations of the costumers are generated randomly and uniformly distributed over an 100×100 area. In each data set for the generated test problems, demands and fixed costs for scenario-1 were drawn uniformly from [0,10000] and [4000, 8000] respectively and then rounded to the nearest integer. Additional scenarios are identified by multiplying scenario-1 to a random number drawn uniformly from interval [0.5, 1.5]. Travel distances or travel times between facilities and customers for scenario-1 were equal to Euclidean distances, and other scenarios are obtained by multiplying random number drawn uniformly from [0.8, 1.3] to the scenario-1 data. Moreover, probability of that scenario s occur is calculated from equation (24).

$$\omega_s = \frac{\sum_{i \in I} W_i^s}{\sum_{s \in S} \sum_{i \in I} W_i^s} \quad \forall s \in S. \tag{24}$$

Considered capacity for each test problem is computed by means of equation (25). In other words, maximum demand in all possible scenarios is determined and then is divided to number of facilities.

$$C = \frac{\max_{\forall s \in S} \left\{ \sum_{i \in I} W_i^s \right\}}{|P|}. \tag{25}$$

5.2 Computational Results

The proposed *p*-SCFLP has not been considered formerly in the literature. Consequently, we cannot provide comparisons with other computational tests. Accordingly, we test the efficiency of algorithms by solving eighteen test problems with various dimensions and comparing obtained results with CPLEX. Following parameters in this experiment were used.

Maximum considered CPU time for all test problems was fixed at 4000 seconds. Ten and fifteen percent of network's nodes are considered as the number of facilities, or *P*. In addition, parameters of the proposed VNS algorithms were set to $K_{\max} = 3,\ B_1 = 15,\ B_2 = 10,\ B_3 = 5$ and the SA parameters were $\alpha = 0.97,\ T_0 = 100,\ TF = 0.1$. Regarding the proposed model, the required parameters *a* and *b* were respectively set to 10^4 and 1 which yields in a linear mixed-integer formulation. To carrying the experimental tests, in this section, we first solve the minmax regret formulation, RCFLP, and then present computational result for the proposed min-expected-cost formulation, p-SCFLP.

In Tables 2 through 5, those columns under "*Test Problems*" which are labeled by "*TP*", "*N*", "*P*", and "*S*" indicate number of test problems, number of network's nodes, number of allowable facilities to be open, and number of scenarios, respectively. Likewise, those columns which are labeled by "*L.B*","*GAP%*", and"*CPU Time(s)*" represent lower bound, gap or relative error (see equation (22)), and elapsed CPU time in seconds, respectively. In addition, the average CPU time requirements in seconds and average gap are listed in the last row of Tables 2-5. Moreover, in these tables best performance in solving each test instance is boldfaced for the better algorithm.

Table 2: Computational results for the generated test instances: RCFLP

Test Instances				CPLEX			VNS Algorithm		SA Algorithm	
TP	N	P	S	L.B	GAP%	CPU Time (s)	GAP%	CPU Time (s)	GAP%	CPU Time (s)
TP1	10	1	10	0.303	0	1.94	0	**0.36**	0	0.86
		2	10	0.116	0	2.06	0	**0.41**	0	1.73
TP2	20	2	10	0.507	0	9.59	0	**1.39**	0	4.40
		3	10	0.094	0	4.21	0	**1.59**	0	2.41
TP3	40	4	10	0.288	0	33.68	0	**0.64**	0	2.64
		6	10	0.018	0	58.17	0	**5.54**	0	32.84
TP4	60	6	10	0.123	0	1058.96	0	27.50	0	**3.25**
		9	10	0.097	**0**	2609.89	0.21	**103.61**	1.12	210.46
TP5	80	8	10	0.329	8.58	4000	**5.46**	**612.12**	11.45	1600.31
		12	10	0.051	29.49	4000	**19.01**	**343.29**	24.36	451.41
Average:					3.81	1177.85	**2.47**	**109.12**	3.73	219.68

5.3 Numerical Experiment for the RCFLP

Note from Table 2 that, for TP1-3, on the average, VNS and SA found the optimal solutions in less than 1.66 and 7.48 seconds, respectively, while CPLEX required 18.28 seconds in order to achieve to the optimal solutions. Likewise, for larger instances (TP4 and 5), the heuristics respectively by consuming 271.63 and 566.36 seconds also outperformed CPLEX which elapsed 2917.21 seconds. All in all, VNS, SA, and CPLEX respectively by elapsing 109.12, 219.68, and 1177.85 seconds obtained solutions that have 2.47, 3.73, and 3.81 percentages gap with the lower bound. As a result, the proposed VNS algorithm has been able to outperform both SA and CPLEX in terms of optimality gap and CPU time requirement. More interestingly that in one case the VNS was able to improve CPLEX's solution by more than 10 percentages.

The proposed algorithmshave been also applied to solve well-known test problems taken from literature. Our heuristics solved most of these instances to optimality within a small fraction of CPLEX's CPU time. On the average, VNS was able to improve solutions optimality gap by 40.04% and reduce CPU time requirement by 30.47% in compare with CPLEX. On the other hand, these values for SA algorithm are respectively 38.67 and 10.37 percentages. Accordingly, the proposed VNS outperforms again both SA and CPLEX.

Table 3: Computational results for the standard test instances: RCFLP

Test Instances				CPLEX			VNS Algorithm		SA Algorithm	
TP	N	P	S	L.B	GAP%	CPU Time (s)	GAP%	CPU Time (s)	GAP%	CPU Time (s)
TP6	49	5	9	0.023	0	8.01	0	**1.28**	0	13.17
		7	9	0.016	0	5.44	0	5.24	0	**2.10**
TP7	55	5	5	0.066	0	606.33	0	**47.62**	0	51.39
		8	5	0.029	0	71.37	0	**30.93**	0	76.46
TP8	88	9	9	0.024	0	106.15	0	**85.39**	0	226.22
		13	9	0.018	0	78.53	0	**37.72**	1.09	117.85
TP9	150	15	9	0.016	33.05	4000	15.79	**2083.67**	**10.61**	3690.75
		22	9	0.010	**8.51**	4000	9.13	**3879.17**	13.79	3745.34
Average:					5.20	1109.48	**3.12**	**771.38**	3.19	990.41

5.4 Numerical Experiment for the p-SCFLP

In this part, we present computational result for the proposed model with min-expected-cost objective function (equations (2 to 9)). To carrying this experiment, the p value for each instance was set to its lower case taken from Tables 2 and 3. In general, since the min-expected-cost formulation is much less complex than minmax formulation, CPLEX could find optimal solution almost for all instances within a small fraction of the considered CPU time. As a result, the algorithms were not able to improve CPLEX's results too much. The next two consecutive tables summarize the computational results of this experiment.

Table 4**:** Computational results for the generated test instances: p-SCFLP

Test Instances				CPLEX			VNS Algorithm		SA Algorithm	
TP	N	P	S	L.B	GAP%	CPU Time (s)	GAP%	CPU Time (s)	GAP%	CPU Time (s)
TP1	10	1	10	1337439.276	0	1.59	0	**0.35**	0	0.317
		2	10	665135.906	0	1.75	0	**0.37**	0	0.71
TP2	20	2	10	2889820.079	0	5.18	0	**2.45**	0	5.85
		3	10	2101115.553	0	3.37	0	**1.75**	0	2.53
TP3	40	4	10	3440772.095	0	16.19	0	**0.67**	0	0.83
		6	10	2706890.175	0	33.56	0	79.82	0	**5.55**
TP4	60	6	10	3321582.368	0	342.41	0	32.529	0	**15.05**
		9	10	2551651.167	0	**162.93**	0.01	195.57	0	233.13
TP5	80	8	10	4735144.137	**0**	2367.63	0.25	**144.04**	0.06	2266.88
		12	10	3305350.598	0	104.80	0	**51.65**	0	101.35
Average:					**0**	303.94	0.03	**50.92**	0.01	263.22

Table 5: Computational results for the standard test instances: p-SCFLP

Test Instances				CPLEX			VNS Algorithm		SA Algorithm	
TP	N	P	S	L.B	GAP%	CPU Time (s)	GAP%	CPU Time (s)	GAP%	CPU Time (s)
TP6	49	5	9	27769.503	0	3.55	0	**0.83**	0	1.38
		7	9	38813.809	0	2.83	0	2.58	0	**1.83**
TP7	55	5	5	14496.678	0	182.69	0	103.08	0	**70.67**
		8	5	14540.330	0	14.23	0.18	**4.89**	0	6.11
TP8	88	9	9	4690.678	0	17.01	0	27.11	0	**6.46**
		13	9	6699.623	0	**10.23**	0	56.85	0	64.22
TP9	150	15	9	15412.387	0.14	4000	**0.02**	**974.24**	0.29	2140.21
		22	9	22366.534	0.16	4000	**0.05**	**2997.89**	0.18	3849.25
Average:					0.05	1028.82	**0.03**	**520.93**	0.06	767.52

Table 4 summarizes performance comparison between algorithms and CPLEX for the generated test instances. As shown in Table 4, VNS and SA, unlike CPLEX, failed to obtain the optimal solutions for 2 and 1 instances, respectively. Moreover, on the average, CPLEX, VNS, and SA for the instances which are solved optimally required 63.61, 21.20, and 16.52 seconds, respectively. However, taking all instances together, the proposed VNS outperforms both SA and CPLEX in terms of CPU time requirement and solution quality.

Table 5 reveals computational results on the standard test problems. For the majority of instances, all approaches found the optimal solutions taking a few minutes. However, the best solutions qualities and fastest performance was achieved by VNS. Note that, the proposed VNS in two cases, TP9, in addition to reducing the CPLEX's CPU time, improved optimality gaps as well. These experimental studies clearly demonstrate the advantages of using the proposed VNS in solving the p-SCFLP.

In order to investigate the convergence speed of proposed solution algorithms, we took TP5, the test problem with 80 nodes and 8 facilities, as a representative and observed algorithms and CPLEX's behavior in reducing optimality gap over time. As a result, whenever the objective function improved, its gap with the lower bound reported in Table 2 and its time were recorded and the results are depicted in Fig. 4. As this plot shows, the proposed initial solution has less optimality gap in compare with the first feasible solution of the CPLEX, roughly 34%. Clearly, the proposed VNS much faster than SA or CPLEX converges to the best solution. Though after approximately 600 seconds the VNS could not improve the solution any further, it has achieved to the closest solution to the lower bound.

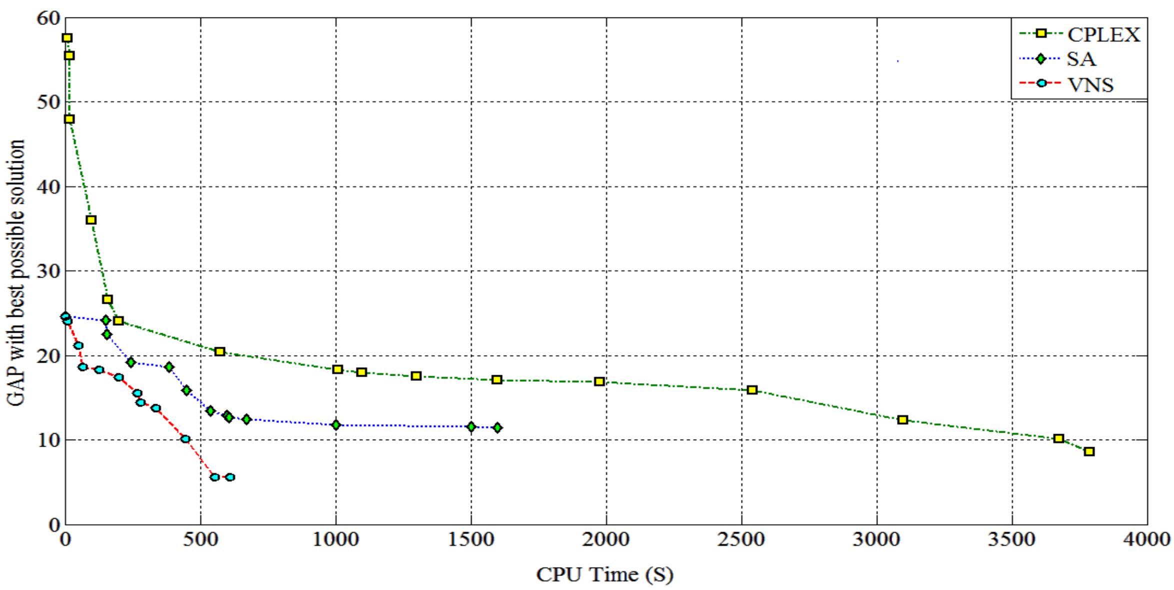

Figure 4: Convergence comparison between CPLEX, VNS and SA to solve TP5 over time

6 Conclusion and Future Research Direction

In this paper an extension of the capacitated facility location problem under uncertainty was investigated. The mathematical formulation was developed with these assumptions that demands and transportations costs are uncertain. In addition, since facilities have hard constraint on the amount of demand they can supply, it is likely that they cannot completely serve all demands. As a result, we developed the model formulation to allow partial satisfaction by introducing a penalty cost as a function of unsupplied demand. In this model, the objective function minimized expected costs while relative regret in each scenario was restricted.In addition, we discussed the minmax formulation in which the worst case is minimized.

The proposed model formulation is a NP-Hard problem and very challenging to solve. As a result, two fix-and-optimize heuristics based on variable neighborhood search and simulated annealing were developed to solve the model. In line with this subject, the problem was split into two sub-problems, location sub-problem and assignment sub-problem. The proposed algorithms iteratively determine the facilities' location and then the assignment sub-problem is solved by CPLEX. Algorithms' performance in terms of optimality gap and CPU time requirement was tested by using a variety of test problems up to 150 nodes. We presented numerical experiments for both minmax regret formulation and min-expected-cost objective function. The results demonstrated that the algorithms besides of simplicity outperform CPLEX in terms of solution quality and computer time requirement. In general, the best performance was achieved by the VNS.

Further attention is also required in the future researches to include additional real assumptions such as disruption in which facilities may randomly fail to service customers or any practical assumption which can be helpfully considered. Another interesting research avenue is to extend the model to take into account accessibility to different classes of facilities with various level of hierarchy. Another line of research would be developing other alternative solution algorithms to solve larger instances. Moreover, developing an efficient local search algorithm might be an appropriate future researches direction.

Acknowledgements

We would like to thank two anonymous referees for their constructive comments and suggestions which improved both the insights and presentation of the paper.

References

[1] Aardal, K., Capacitated facility location: separation algorithm and computational experience, *Mathematical Programming*, vol.81, pp.149–175, 1998.

[2] Averback, I., The minmax relative regret median problem on networks, *INFORMS Journal on Computing*, vol.17, no.4, pp.451–461, 2005.

[3] Barahona, F., and F.A. Chudak, Near-optimal solutions to large-scale facility location problems, *Discrete Optimization*, vol.2, pp.35–50, 2005.

[4] Beasley, J.E., Lagrangian heuristics for location problems, *European Journal of Operational Research*, vol.65, pp.383–399, 1993.

[5] Brandeau, M.L., and S.S. Chiu, An overview of representative problems in location research, *Management Science*, vol.35, no.6, pp.645–674, 1989.

[6] Brimberg, J., and N. Mladenović, A variable neighborhood algorithm for solving the continuous location-allocation problem, *Studies in Locational Analysis*, vol.10, pp.1–12, 1996.

[7] Cornuejols, G., R. Sridharan, and J.M. Thizy, A comparison of heuristics and relaxations for the capacitated plant location problem, *European Journal of Operational Research*, vol.50, pp.280–297, 1991.

[8] Cortinhal, M.J., and M.E. Captivo, Upper and lower bounds for the single source capacitated location problem, *European Journal of Operational Research*, vol.151, pp.333–351, 2003.

[9] Daskin, M.S., *Network and Discrete Location: Models, Algorithms, and Applications*, Wiley, New York, 1995.

[10] Drezner, Z., and H.W. Hamacher, *Facility Location: Theory and Algorithms*, Springer, 2001.

[11] Ferris, M.C., *MATLAB and GAMS: Interfacing Optimization and Visualization Software,* University of Wisconsin, 2005.

[12] Fleszar, K., and K.S. Hindi, An effective VNS for the capacitated p-median problem, *European Journal of Operational Research*, vol.191, pp.612–622, 2008.

[13] Ghezavati, V.R., M. Saidi-Mehrabad, and S.J. Sadjadi, A robust approach to location-allocation problem under uncertainty, *Journal of Uncertain Systems*, vol.3, no.2, pp.131–136, 2009.

[14] Hansen, P., N. Mladenović, and J.A.M. Pérez, Variable neighbourhood search: methods and applications, *Annals of Operations Research*, vol.175, pp.367–407, 2010.

[15] Hindi, K.S., and K. Pienkosz, Efficient solution of large scale, single source, capacitated plant location problem, *Journal of the Operational Research Society*, vol.50, pp.268–274, 1999.

[16] Jabal-Ameli, M.S., N. Azad, and A. Rastpour, Designing a supply chain network model with uncertain demands and lead times, *Journal of Uncertain Systems*, vol.3, no.2, pp.123–130, 2009.

[17] Jacobsen, S.K., Heuristics for the capacitated plant location model, *European Journal of Operational Research*, vol.12, pp.253–261, 1983.

[18] Jayaraman, V., and A. Ross, A simulated annealing methodology to distribution network design and management, *European Journal of Operational Research*, vol.144, pp.629–645, 2003.

[19] Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, Optimization by simulated annealing, *Science*, vol.220, pp.671–680, 1983.

[20] Korupolu, M., C. Plaxton, and R. Rajaraman, Analysis of a local search heuristic for facility location problems, *Journal of Algorithms*, vol.37, pp.146–188, 2000.

[21] Kouvelis, P., and G. Yu, *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, 2000.

[22] Kuehn, A.A., and M.J. Hamburger, A heuristic program for locating warehouses, *Management Science*, vol.9, pp.643–666, 1963.

[23] Lorena, L.A.N., and E.L.F. Senne, Improving traditional subgradient scheme for Lagrangean relaxation: an application to location problems, *International Journal of Mathematical Algorithms*, vol.1, pp.133–151, 1999.

[24] Marianov, V., and D. Serra, Location problems in the public sector, *Facility Location: Applications and Theory*, edited by Z. Drezner and H.W. Hamacher, Springer, Berlin, 2002.

[25] Nauss, R.M., An improved algorithm for the capacitated facility location problem, *Journal of the Operational Research Society*, vol.29, pp.1195–1201, 1978.

[26] Owen, S.H., and M.S. Daskin, Strategic facility location: a review, *European Journal of Operational Research*, vol.111, pp.423–447, 1998.

[27] Pahlavani, A., and M. Saidi-Mehrabad, A fuzzy multi-objective programming for optimization of hierarchical service centre locations, *Journal of Uncertain Systems*, vol.5, no.3, pp.202–226, 2011.

[28] Sa, G., Branch and bound and approximate solutions to the capacitated plant location problem, *Operations Research*, vol.17, pp.1005–1016, 1969.

[29] Sahinidis, N.V., Optimization under uncertainty: state-of-the-art and opportunities, *Computers & Chemical Engineering*, vol.28, pp.971–983, 2004.

[30] Snyder, L.V., M.S. Daskin, and C.P. Teo, The stochastic location model with risk pooling, *European Journal of Operational Research*, vol.179, pp.1221–1238, 2007.

[31] Snyder, L.V., Facility location under uncertainty: a review, *IIE Transactions*, vol.38, pp.537–554, 2006.

[32] Snyder, L.V., and M.S. Daskin, Stochastic p-robust location problems, *IIE Transactions*, vol.38, pp.971–985, 2006.

[33] Sridharan, R., Invited review the capacitated plant location problem, *European Journal of Operational Research*, vol.87, pp.203–213, 1995.

[34] Sun, M., A tabu search heuristic procedure for the capacitated facility location problem, *Journal of Heuristics*, vol.18, pp.91–118, 2012.

[35] Weaver, J.R., and R.L. Church, Computational procedures for location problems on stochastic networks, *Transportation Science*, vol.17, pp.168–180, 1983.

[36] Yu, V.F., S.-W. Lin, W. Lee, and C.-J. Ting, A simulated annealing heuristic for the capacitated location routing problem, *Computers and Industrial Engineering*, vol.58, no.2, pp.288–299, 2010.