Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A hybrid algorithm for stochastic single-source capacitated facility location problem with service level requirements

Hosseinali Salemi[a*]

School of industrial and information engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133, Milano, Italy

CHRONICLE

Article history:
Received July 21 2015
Received in Revised Format
August 16 2015
Accepted October 8 2015
Available online
October 8 2015

Keywords:
Supply chain management
Lagrangian heuristic
Ant colony optimization
Genetic optimization
Service level
Stochastic facility location problem Poisson distribution
Hybrid Algorithm

ABSTRACT

Facility location models are observed in many diverse areas such as communication networks, transportation, and distribution systems planning. They play significant role in supply chain and operations management and are one of the main well-known topics in strategic agenda of contemporary manufacturing and service companies accompanied by long-lasting effects. We define a new approach for solving stochastic single source capacitated facility location problem (SSSCFLP). Customers with stochastic demand are assigned to set of capacitated facilities that are selected to serve them. It is demonstrated that problem can be transformed to deterministic Single Source Capacitated Facility Location Problem (SSCFLP) for Poisson demand distribution. A hybrid algorithm which combines Lagrangian heuristic with adjusted mixture of Ant colony and Genetic optimization is proposed to find lower and upper bounds for this problem. Computational results of various instances with distinct properties indicate that proposed solving approach is efficient.

1. Introduction

A segment of supply chain management is dedicated to facility location problems (FLP). Given a set of potential locations of facilities (e.g. plants, warehouses) and a set of demand points, facility location problem is to find locations of facilities in a way that total cost of opening and allocation of customers for satisfying their demand is minimized. These problems may be seen as the most critical and significant decisions in building an efficient supply chain since they are fixed and difficult to change even in the intermediate term. The location of a multibillion-dollar facility cannot be changed as a result of changes in customer demands, raw material price or transportation rates. There are different categories of facility location problem relying on restrictions assumption. One concern in FLP is the capacity of each facility for serving its assigned demand points. While in the uncapacitated facility location problem, capacity of each facility is assumed to be infinite, in capacitated facility location problem (CFLP), each facility has a finite capacity. Single Source Capacitated Facility Location Problem (SSCFLP) which is proven to

* Corresponding author. Tel: +39-3207761926
E-mail: hosseinali.salemi@mail.polimi.it (H. Salemi)

doi: 10.5267/j.ijiec.2015.10.001

belong to category of NP-hard problems, is a subset of CFLP where each customer receives all its demand from just one opened facility. SSCFLP is the opposite point of multi-sourcing capacitated problems. In real world, demand of customers are not pre-determined and definite, but stochastic and indecisive. Hence, facilities should have sufficient capacity to cope with uncertainties with a degree of reliability level.

Following model is Stochastic Single Source Capacitated Facility Location Problem (SSSFLP) that determines which facilities should be opened and which demand points are allocated to each of them in a way that total cost is minimized:

Notation

I $\{1,2,\dots,m\}$ $set\ of\ candidate\ facilities$

J $\{1,2,\dots,n\}$ $set\ of\ demand\ points$

Decision variables

y_i $determines\ whether\ facility\ i\ is\ opened\ or\ not$

X_{ij} $determines\ whether\ demand\ point\ j\ is\ assigned\ to\ facility\ i\ or\ not$

Formulation

$$(\mathfrak{S}) \quad z = min \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} X_{ij}$$

Subject to:

$$\sum_{i=1}^{m} X_{ij} = 1 \qquad j = 1,2,\dots,\mathrm{n} \tag{1}$$

$$prob\ (\sum_{j=1}^{n} a_j X_{ij} \le Q_i y_i) \ \ge\ \alpha_i \qquad i = 1,2,\dots,m \tag{2}$$

$$X_{ij} \ \le y_i \qquad i = 1,2,\dots,m \quad j = 1,2,\dots,\mathrm{n} \tag{3}$$

$$y_i = \begin{cases} 1, & if\ facility\ i\ is\ opened \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m \tag{4}$$

$$X_{ij} = \begin{cases} 1, & if\ facility\ i\ is\ assigned\ to\ demand\ point\ j \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m \quad j = 1,2,\dots,\mathrm{n} \tag{5}$$

The decision is about selecting a set of facilities among m candidates and assignment of each demand point j $(j = 1,2,\dots,n)$ to exactly one opened facility i. The objective function minimizes opened facilities fixed costs f_i and serving costs c_{ij} related to assignment of each customer j to a unique facility i. Constraint (1) guarantees that each demand point is exactly assigned to just one facility. In real world, customers' demands are not deterministic and facilities should have sufficient capacity to cope with requested demand from them. Instead of considering stochastic demand for each individual customer j, cumulative demand assigned to each facility i is considered probabilistic. Reliability level or capability of facility i is represented by α_i $(0 < \alpha_i < 1)$. The higher the reliability level of facility i, the lower the probability of not being capable of satisfying customer demands assigned to that. a_j and Q_i represents orders requested by demand point j and orders which facility i can serve to its assigned demand points, respectively. Constraint (2), with probability higher than α_i, stipulates that demands which facility i is capable of serving to its allocated customers is equal or higher than sum of orders that demand points assigned to that facility requested. Constraint (3) certifies that assignment of demand points to a facility takes place just if that facility is opened. Constraints (4) and (5) show that y_i and X_{ij} are binary variables that can be zero or one.

The paper is organized in 6 sections. Section 2 reviews the literature about different methods of solving facility location problems. Section 3 examines the case of Poisson demand distribution. Section 4 introduces the novel and state-of-the-art hybrid algorithm to solve SSSCFLP. In section 5 computational results are provided. Conclusion is presented in section 6.

2. Literature review

Different approaches for solving SSCFLP is extant in literature. Neebe and Rao (1983) formulate SSCLP as a set partitioning problem and solve it by a column-generating branch and bound procedure. Bounds are obtained by using linear relaxation. Due to the high probability of having integer solutions with the linear program, they observe that the tree search is not very large.

The most successful approach known to be is Lagrangian heuristic. The basis of these heuristics is a combination of Lagrangian relaxation and methods for solving the Lagrangian dual problem with the aim of finding lower and upper bounds of optimal solution. Beasley (1993) introduces a framework to develop Lagrangian heuristics based on Lagrangian relaxation and subgradient optimization for location problems in which both capacity and demand points constraints are relaxed and dualized. His work provides efficient solution for different types of location problems like uncapacitated, p-median, capacitated and single source capacitated facility location problems. Holmberg et al. (1999) propose Lagrangian heuristic involving Lagrangian relaxation, subgradient optimization and primal heuristic within a branch-and-bound framework to solve SSCFLP with high quality and robustness. The primal heuristic is based on solving a sequence of related matching problems and is incorporated into Lagrangian heuristic. Tragantalerngsak et al. (2000) present a Lagrangian relaxation-based branch-and-bound algorithm for solving a particular type of FLP in which two echelons of facilities exist. Only one facility in the first echelon can serve each facility of the second echelon which has a limited capacity and each demand point receives its orders from just one facility in the second echelon. Klincewicz and Luss. (1986) use Lagrangian heuristic to solve SSCFLP by relaxing capacity constraints, obtaining uncapacitated facility location as a subproblem and using dual ascent algorithm. A heuristic based on cost differentials is applied to decrease cost of solution obtained in Lagrangian phase. Barcelo and Casanovas (1984) present a Lagrangian heuristic solution in which customer assignment constraints are relaxed and series of knapsack problems are generated. The proposed Lagrangian heuristic involves two sections of facility location selection and demand points allocation. In assignment of customers to opened facilities, regret heuristic is used.

Metaheuristics have also been considered as efficient techniques. Chen and Ting (2008) describe multiple ant colony system (MACS) and a hybrid algorithm which combines a Lagrangian heuristic and ant colony system to solve SSCFLP. Two ant colony systems, one for location selection and one for customer allocation are embedded in MACS. Ahuja et al. (2004) propose a very large scale neighborhood (VLSN) search algorithm for SSCFLP. The neighborhood structure are induced by customer multi-exchange and by facility moves. Both single-customer multi-exchanges and multi-customer multi-exchanges are considered. Cortinhal and Captivo (2003) apply Lagrangian heuristic followed by search methods that is local search and tabu search metaheuristic to obtain upper bounds of SSCFLP. According to their computational results, tabu search has better performance in comparison to local search. Hindi and Pienkosz (1999) develop a heuristic capable of solving large instances of SSCFLP. It is a combination of Lagrangian heuristic and restricted neighborhood search. Lagrangian relaxation combined with subgradient procedure yields the lower bounds and heuristic search method which is the restricted neighborhood (tabu-like) search provides upper bounds.

The objective of this work is to develop a method and algorithm to provide good solutions for Stochastic Single Source Capacitated Facility Location Problem (SSSCFLP). Poisson distribution demand is considered and its equivalent deterministic model and proposed solution algorithm is described.

3. Poisson demand

We consider Poisson demand distribution where demand node j has mean λ_j $(j \in J)$. This distribution function for customer demand can be observed in medical, emergency or telecommunication services. Now, we try to change the stochastic problem ($\mathfrak{S}$) to deterministic one as done in C.K.Y. (Lin, 2009). Since sum of a set of Poisson random variables is a Poisson variable, $\sum_{j=1}^{n} a_j X_{ij}$, which is the total demand assigned to facility i can be modeled by a Poisson variable of mean $\sum_{j=1}^{n} \lambda_j X_{ij}$. So, Eq. (2) in ($\mathfrak{S}$) , can be rewritten as:

$$\sum_{k=0}^{Q_i y_i} e^{-(\sum_{j=1}^{n} \lambda_j X_{ij})} \frac{(\sum_{j=1}^{n} \lambda_j X_{ij})^k}{k!} \geq \alpha_i \quad i = 1,2,\dots,m \tag{6}$$

Assume we have a Poisson variable x whose mean is the largest value satisfying condition $prob\ (x \leq Q_i) \geq \alpha_i$. We denote mean of x by b_i and vary it until $prob\ (x \leq Q_i) \cong \alpha_i$ is obtained. b_i, which is a function of Q_i and α_i, can be easily found by try and error method. In this way we can rewrite Eq. (6) as:

$$\sum_{j=1}^{n} \lambda_j X_{ij} \leq b_i y_i \qquad i = 1,2,\dots,m \tag{7}$$

If we replace Eq. (7) with Eq. (2), stochastic model is changed to a deterministic one:

$$(\mathfrak{D}) \quad z = min \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} X_{ij}$$

subject to

$$\sum_{i=1}^{m} X_{ij} = 1 \qquad \mathrm{j} = 1,2,\dots,\mathrm{n}$$

$$\sum_{j=1}^{n} \lambda_j X_{ij} \leq b_i y_i \qquad i = 1,2,\dots,m$$

$$X_{ij} \leq y_i \qquad i = 1,2,\dots,m \qquad \mathrm{j} = 1,2,\dots,\mathrm{n}$$

$$y_i = \begin{cases} 1, & if\ facility\ i\ is\ opened \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m$$

$$X_{ij} = \begin{cases} 1, & if\ facility\ i\ is\ assigned\ to\ demand\ point\ j \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m \qquad \mathrm{j} = 1,2,\dots,\mathrm{n}$$

If we solve ($\mathfrak{D}$), it will guarantee that constraint $\sum_{j=1}^{n} \lambda_j X_{ij} \leq b_i y_i$ is satisfied and since b_i is the largest value satisfying $prob\ (x \leq Q_i) \cong \alpha_i$, it assures that with probability higher than α_i , condition $\sum_{j=1}^{n} a_j X_{ij} \leq Q_i y_i$ is met which is exactly Eq. (2) of problem ($\mathfrak{S}$).

4. Hybrid algorithm (Combination of Lagrangian Heuristic and adjusted mixture of Ant colony and Genetic Metaheuristics optimization algorithm)

Proposed algorithm combines well-known Lagrangian heuristic with adjusted mixture of Ant colony and Genetic Metahuristics to solve ($\mathfrak{D}$)**.** The main idea behind this algorithm is to obtain a lower bound through Lagrangian relaxation which is enclosed by subgradient optimization method with the aim of generating a sequence of Lagrangian multipliers and acquiring highest possible lower bound. At each iteration of subgradient optimization the adjusted mixture of Ant colony and Genetic Metahuristics are applied to construct a feasible solution and possibly generate better upper bounds. These steps are provided in three phases as follows:

Phase 1. Lagrangian Relaxation: with Lagrangian relaxation, lower bounds on the optimal objective function of ($\mathfrak{D}$) will be produced.

Phase 2. Subgradient optimization: in this phase Lagrangian dual is solved, lower bounds will be improved and Lagrangian multiplier is updated.

Phase 3. Adjusted mixture of Ant colony and Genetic Metaheuristics optimization: In this phase we find feasible solutions as well as possibly update upper bounds on the objective function of ($\mathfrak{D}$).

In the following each phase will be described in detail:

Phase1. Lagrangian Relaxation

Lagrangian relaxation is a technique to obtain a relaxed problem, called Lagrangian subproblem, which is easier to be solved than original problem. The basic idea is to remove or relax set of problem constraints and put them into objective function assigned with weights (Lagrangian multiplier). Each weight is the representation of a penalty which is added to solution that does not satisfy the set of relaxed constraints. Since problem ($\mathfrak{D}$) is a minimization problem, the objective value of its relaxed problem provides a lower bound for it.

We relax constraint $\sum_{i=1}^{m} X_{ij} = 1$ $(j = 1,2,\dots,n)$ of ($\mathfrak{D}$) with associated Lagrangian multiplier μ_j $(\mu_j \geq 0)$:

$$(\mathfrak{L}) \quad z(\mu) = min \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}X_{ij} + \sum_{j=1}^{n} \mu_j (1 - \sum_{i=1}^{m} X_{ij})$$

subject to

$$\sum_{j=1}^{n} \lambda_j X_{ij} \leq b_i y_i \qquad i = 1,2,\dots,m$$

$$X_{ij} \leq y_i \qquad i = 1,2,\dots,m \quad j = 1,2,\dots,n$$

$$y_i = \begin{cases} 1, & if\ facility\ i\ is\ opened \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m$$

$$X_{ij} = \begin{cases} 1, & if\ facility\ i\ is\ assigned\ to\ demand\ point\ j \\ 0, & otherwise \end{cases} \qquad i = 1,2,\dots,m \quad j = 1,2,\dots,n$$

In ($\mathfrak{L}$), which is the relaxed problem of ($\mathfrak{D}$), a starting point for Lagrangian multiplier (${\mu_j}^{(1)}$) is chosen. If we assume $y_i = 1$ $(i = 1,2,\dots,m)$, for each facility i we have the following knapsack problem ($\mathcal{K}$). In other words, ($\mathfrak{L}$) is considered as m separated knapsack problems (one problem per each facility i). It should be mentioned that if $c_{ij} - \mu_j \geq 0$, then X_{ij} can be removed from ($\mathcal{K}$) because the optimal value of X_{ij} will be zero.

$$(\boldsymbol{\mathcal{K}}) \quad k = min\sum_{j=1}^{n}(c_{ij}-\mu_j)X_{ij}$$

subject to

$$\sum_{j=1}^{n}\lambda_j X_{ij} \ \le b_i y_i \qquad\qquad i = 1,2,\dots,m$$

$$X_{ij} = \begin{cases}1, \ if\ facility\ i\ is\ assigned\ to\ demand\ point\ j \\ 0, \ otherwise\end{cases} \qquad i = 1,2,\dots,m \quad j \ = \ 1,2,\dots,n$$

After solving ($\boldsymbol{\mathcal{K}}$), the term $f_i + \sum_{j=1}^{n}(c_{ij}-\mu_j)X_{ij}$ which can be noticed as reduced cost for y_i is calculated for each and every facility. If it is lower than zero, then $y_i = 1$ and X_{ij} are the quantities that have been obtained from knapsack problem. Otherwise, both y_i and X_{ij} will be zero:

$$if\ f_i + \sum_{j=1}^{n}(c_{ij}-\mu_j{}^{(1)})X_{ij} \ < 0 \qquad y_i = 1\ ,X_{ij} = Quantities\ obtained\ from\ (\mathcal{K})$$

$$if\ f_i + \sum_{j=1}^{n}(c_{ij}-\mu_j{}^{(1)})X_{ij} \ > 0 \qquad y_i = 0\ ,X_{ij} = 0$$

Proof

We know that $f_i \ge 0$ and $\sum_{j=1}^{n} c_{ij}X_{ij} \ge 0$. So, if $f_i + \sum_{j=1}^{n} c_{ij}X_{ij} - \ \sum_{j=1}^{n}\mu_j X_{ij} < 0$ then $\sum_{j=1}^{n}\mu_j X_{ij} > 0$. We also know that Lagrangian multiplier $\mu_j \ \ge 0$. Therefore $X_{ij} \ > 0$. X_{ij} is a binary variable which can be zero or one and if it is more than zero it has to be one. Being $X_{ij} = 1$ indicates that facility i is opened so $y_i = 1$.

If $f_i + \sum_{j=1}^{n} c_{ij}X_{ij} - \ \sum_{j=1}^{n}\mu_j X_{ij} \ > 0$, it is not beneficial to open facility i so $y_i = 0$ and consequently $X_{ij} = 0$.

Lower bound

The optimal value of objective function ($\mathfrak{L}$) which is the lower bound will be:

$$z(\mu) = \sum_{i=1}^{m} \ (\ f_i\widehat{y_i} + \sum_{j=1}^{n}(c_{ij}-\mu_j)\hat{X}_{ij}\ ) + \ \sum_{j=1}^{n}\mu_j$$

where $\widehat{y_i}$ and $\hat{X}_{ij}$ are values obtained in the way which was explained.

Phase 2. Subgradient optimization

Lagrangian dual problem is:

$(\boldsymbol{\mathcal{L}}) \quad w = \max\ z(\mu)$

An efficient method to solve the dual problem with non-differentiable function and find the optimal Lagrangian multiplier is subgradient optimization. Our subgradient method uses the vector $\hbar_j = 1 - \sum_{i=1}^{m}\hat{X}_{ij}$ $(\forall j)$ for optimal Lagrangian solutions as though it is the gradient of $z(\mu)$.

We choose starting points $k = 1, \varepsilon_1 > 0,\ \mu^{(1)}$ and obtain $(\hat{X}, \hat{y})$ as well as $z(\mu^{(k)})$ as described in phase 1. Moreover, We denote the best known upper bound and lower bound for objective function of $(\mathfrak{L})$ with $\bar{v}$ and $\underline{v}$ respectively. We initialize $\bar{v} = \sum_{j=1}^{n} (\max_i c_{ij}) + \sum_{i=1}^{m} f_i$ and $\underline{v} = -\infty$.

If $z(\mu^{(k)}) > \underline{v}$ then we let $\underline{v} = z(\mu^{(k)})$. Afterwards we try to modify $(\hat{X}, \hat{y})$ into a feasible solution and possibly update $\bar{v}$ with adjusted mixture of Ant colony and Genetic Metaheuristics optimization which will be completely explained in phase 3.

These steps are repeated until stopping criteria $|\hbar^k| \le \varepsilon_1$ or maximal number of iterations allowed are met. Then $\bar{v}$ is accepted as the best solution to $(\mathfrak{D})$. If stopping criteria are not met, we let $k = k + 1$ and solve $(\mathfrak{L})$ again yielding new $(\hat{X}, \hat{y})$ and $z(\mu^{(k)})$ by updating $\mu^{(k)}$ as:

$$\mu^{(k+1)} = \max\{\mu^{(k)} + t^{(k)}\hbar^{(k)}, 0\}$$

where $t^{(k)}$ is the step size and can be calculated by:

$$t^{(k)} = \gamma \frac{\bar{v} - z(\mu^{(k)})}{|\hbar^k|^2}$$

γ is a scalar choosing between 0 and 2. We let $\gamma = 2$ and if no improvements in $\underline{v}$ is obtained in three consecutive iterations, γ is halved. In addition, γ is restarted to its initial value in every 100 iterations.

Phase 3. Adjusted mixture of Ant colony and Genetic Metaheuristics optimization

At each iteration of subgradient optimization, we use an adjusted mixture of Ant colony and Genetic metaheuristics optimization algorithm to modify $(\hat{X}, \hat{y})$ into a feasible solution and possibly update $\bar{v}$. In fact, output of $(\mathfrak{L})$ will yield strong lower bounds and very good starting points for our proposed algorithm. With this methodology, we want to assign demand points with no deliveries to facilities in order to update upper bound of the objective function of $(\mathfrak{D})$. The artificial ants carry out a randomized construction heuristic which makes probabilistic decisions as a function of artificial pheromone trails and available heuristic information based on the output data obtained from solving $(\mathfrak{L})$. Each solution constructed by ants is considered as an initial chromosome in genetic algorithm. In genetic algorithm, we evaluate each chromosome based on fitness function and select bests of them for mating pool. Then we apply cross over and mutation operators on chromosomes in mating pool to create offspring with possibly better fitness function value. Previous population will be replaced by resulting mating pool and assessed according to fitness function.

This phase includes three steps as follows:

Step 1. Removing deliveries to demand points receiving their demand from more than one facility.

Step 2. Implementing Ant colony optimization

Step 3. Implementing Genetic optimization

In the following we will describe each step in details:

Step 1. Removing deliveries to demand points receiving their demand from more than one facility

In phase 1, during solving m separated knapsack problems, it may happen that more than one facility is allocated to a customer. For each demand point with more than one delivery from different facilities, we remove deliveries from facilities with the highest assignment costs.

Step 2. Implementing Ant colony optimization

In this step we revise the Multiple Ant Colony System (MACS) (Chen & Ting, 2008) in a way to prepare good initial chromosomes for Genetic optimization. The aim of ant colony optimization is feeding step 3 by producing chromosomes which guarantee that no facility with lower remained capacity than a customer's demand is assigned to that. In ant colony optimization procedure, we have three main stages. In the first stage we should select number of facilities from the candidate sites and in the second stage we should assign demand points with no deliveries to the selected facilities. Once these stages are fulfilled by an ant, pheromones are updated and next ant constructs its solution. In real world, at first, ants seeking for food moon and test different paths and once it is found they produce and release pheromone trails affecting the behavior of other ants searching for food later in a manner that random paths will not be possibly chosen, but according to the pheromone trails. Thus, here, when an artificial ant finds a good path for assignment of facilities to demand points, other ants are likely to mimic the behavior and follow that path which results in generating good chromosomes.

Stage 1. Selecting facilities among candidates

The heuristic information about choosing facilities is based on the ratio of remained capacity of facility i which can serve to its assigned demand points to its fixed cost $(\acute{b}_\imath/f_i)$. The pheromone level of each candidate site i is the required pheromone information in this stage. In order to let each ant h selects different number of facilities, we generate a number according to $U[0,r]$ which is a discrete uniform distribution where r is an integer number in $[0,m]$ and must be initialized. We denote set of demand points that have not been assigned to any facilities in previous phases by $\mathcal{J}$. The number of facilities $\mathfrak{N}^h$selected among m candidates by ant h is calculated as:

$$if \quad \left[\frac{\sum_{j\in\mathcal{J}}\lambda_j}{\sum_{i=1}^{m}\acute{b}_\imath/m}\right]+U[0,r]\ \geq m \qquad \mathfrak{N}^h=m$$

$$if \quad \left[\frac{\sum_{j\in\mathcal{J}}\lambda_j}{\sum_{i=1}^{m}\acute{b}_\imath/m}\right]+\ U[0,r]\ < m \qquad \mathfrak{N}^h=\left[\frac{\sum_{j\in\mathcal{J}}\lambda_j}{\sum_{i=1}^{m}\acute{b}_\imath/m}\right]+U[0,r]$$

$\acute{b}_\imath$ is the difference between b_i and number of orders that facility i served to different demand points j in solution of $(\boldsymbol{\mathfrak{L}})$. It is obvious if $\mathfrak{N}^h=m$, there is no extra effort required to choose some facilities among m candidates since all of them have to be chosen. In the alternative case, where all of the candidates are not going to be chosen, for knowing which $\mathfrak{N}^h$ facilities to be selected, we have to know τ_i^hwhich is the pheromone level of facility i related to ant h. We initialize matrix τ_i^0 according to each facility fixed cost. $(\tau_i^0=\frac{1}{f_i})$. $\mathfrak{N}^h$ locations are chosen by ant h according to the following equation:

$$P_i^h=\frac{(\tau_i^h)(\psi_i^h)^\beta}{\sum_i(\tau_i^h)(\psi_i^h)^\beta}$$

Facility i is chosen according to P_i^h probability distribution. ψ_i^h is calculated as $\acute{b}_\imath/f_i$. β is a parameter that considers the interrelated influence of τ_i^h on ψ_i^h. (β>0)

Stage 2. Assigning demand points to selected facilities

Now, we assign demand points without any deliveries to selected facilities according to following equation:

$$P_{ij}^h=\frac{(\zeta_{ij}^h)(\varrho_{ij}^h)^\delta}{\sum_{i\in S_h,j\in\mathcal{J}}(\zeta_{ij}^h)(\varrho_{ij}^h)^\delta}$$

Customer j is allocated to facility i according to P_{ij}^h probability distribution. S_h is the set of facilities that ant h selected in stage 1. Matrix ζ_{ij}^h is the pheromone level between facility i and demand point j. ζ_{ij}^h is initialized and updated once an ant constructed its solution. δ is a parameter that considers the interrelated influence of ζ_{ij}^h on ϱ_{ij}^h. (δ>0). ϱ_{ij} is calculated by the following equations:

$$\varrho_{ij}^{h} = \frac{1}{c_{ij}} \qquad if\ \lambda_j \ \leq\ \acute{b_\imath}$$

$$\varrho_{ij}^{h} = \frac{1}{c_{ij}} \times \frac{1}{\varpi(\lambda_j - \acute{b_\imath})} \qquad if\ \lambda_j\ >\ \acute{b_\imath}$$

where ϖ is a pre-specified parameter.

Stage 3. Updating pheromones

After an ant constructs its solution, pheromones must be updated. Both τ_i and ζ_{ij} matrixes are updated with the aim that solution constructed by next ant becomes better. Updating equations are as:

$$\tau_i^{h+1} = (1-\theta)\tau_i^h \qquad if\ facility\ i\ \in\ \Psi$$

$$\zeta_{ij}^{h+1} = \left(1-\acute{\theta}\right)\zeta_{ij}^{h} \qquad if\ arc\ (ij)\ \in\ \Psi\ \text{AND}\ \lambda_j\ \leq\ \acute{b_\imath}$$

$$\zeta_{ij}^{h+1} = \frac{\left(1-\acute{\theta}\right)\zeta_{ij}^{h}}{2^h} \qquad if\ arc\ (ij)\ \in\ \Psi\ \text{AND}\ \lambda_j\ >\ \acute{b_\imath}$$

$$\zeta_{ij}^{h+1} = \frac{\zeta_{ij}^{h}}{2^h} \qquad if\ arc\ (ij) \notin\ \Psi\ \text{AND}\ \lambda_j\ >\ \acute{b_\imath}$$

where Ψ is the solution constructed by ant h involving facilities and assignments. θ and $\acute{\theta}$ which must be initialized are pheromone decay parameters. $(\theta, \acute{\theta}\ \in\ [0,1])$.

Now, if $h <$ ð (ð is number of ants), we let $h = h + 1$ and repeat stages 1, 2 and updating pheromones. Otherwise, we will stop and go to step3.

Step 3. Implementing Genetic optimization

Considering constructed solutions by ants in step 2 as an initial points, in this step we try to have feasible solutions as well as improving them if possible. We consider each solution constructed by ants in previous step as an initial chromosome in our genetic optimization algorithm. As the number of ants is ð, the number of initial chromosomes will be ð too. The basic idea is to make evolution to set of solutions developed in last step toward solutions with lower degree of infeasibility. Infeasibility degree of a solution is defined as total sum of difference of each facility remained capacity and total demands assigned to that. If infeasibility degree of a solution is zero, it is considered feasible.

Genetic Representation

Each chromosome consists of some genes that each of them shows which facility is assigned to which demand point. Number of genes in each chromosome equals to number of elements of set $\mathcal{J}$. For example if customers $\{2,7,9,10\}\ \in\ \mathcal{J}$ are not assigned to any facility in previous phases, each chromosome in genetic optimization have four genes. $\langle 1,5,4,1\rangle$ indicates that demand point 2 and 10 are assigned to facility 1, demand point 7 is assigned to facility 5 and demand point 9 is assigned to facility 4.

In the following, cumulative fitness function is calculated for each initial chromosome. By applying selection criteria, some of best chromosomes are selected. Then crossover and mutation operators are implemented in order to obtain better offspring from the best chromosomes. When we find a feasible solution another mutation operator is applied to possibly makes it better. This procedure is elaborated in details later.

Fitness Function

Fitness function or evaluation function gives a value to each chromosome, indicating its merit in terms of problem objective. It will give an assessment of available chromosomes and a scale for selection.

Cumulative fitness function is calculated as follow:

$$\mathcal{F}_c = (A + B + C)^{-1}$$

$$A = \sum c_{ij} \quad for\ each\ ij\ if\ X_{ij} = 1\ and\ j \in \mathcal{J}$$

$$B = \sum f_i \quad for\ each\ i\ if\ it\ was\ not\ opened\ in\ previous\ phases$$

$$C = \frac{\sum_{j \in \mho} c_{ij} \times \left(\acute{b}_\iota - \sum_{j \in \mho} \lambda_j \right)^{\chi}}{\sum_{j \in \mho} \lambda_j} \quad for\ each\ i$$

where $\mho$ is set of demand points j which is assigned to facility i and x is a parameter that should be initialized.

Selection

After each generation, a number of chromosomes from available population are selected according to selection criterion to breed a new generation likely with a better evaluation function value. The key idea is to give preference to better chromosomes, allowing them to pass on their genes to the next generation. We calculate the cumulative fitness function of all chromosomes and name the cumulative fitness function of last chromosome $\mathscr{q}$. Random numbers are yielded in $(0, \mathscr{q})$. The chromosome that has been placed in the related interval according to cumulative fitness functions is chosen. The higher the fitness function value of a chromosome, the higher the probability of that chromosome to be chosen.

Elitism

Since in producing new generations, there is a risk to lose the best chromosomes, in each generation we keep a copy from the best, top ranked value, fittest chromosomes without any change to the next generation. This will guarantee the existence of them till end of the process and ensure that quality of solutions will not be decreased during breeding new chromosomes. These elitist candidates are eligible for selection during next generations.

Crossover Operator

Crossover operators are used to produce new generations by using chromosomes selected from previous generation and elitism. To generate new set of chromosomes, a pair of parent is selected for mating and breeding a child solution which typically shares many of the characteristics of its parents. Our proposed crossover operator chooses two chromosomes as parents among candidate chromosomes. We name them father1 and father2. $father1[g]$ represents the gth gene of chromosome $father1$. We initialize offspring chromosome as:

$$if\ father1[g] = father2[g] \quad offspring[g] = father1[g]$$

$$if\ father1[g] \neq father2[g] \quad offspring[g] = 0$$

while there is a g that $offspr[g]$=0, we randomly choose a position t that $offspring[t]$=0 and calculate degree of infeasibility. If $offs[t]$=$father1[t]$ yields lower degree of infeasibility or same degree and a lower cij, then $offspring[t]$=$father1[t]$. Otherwise, $offs[t]$=$father2[t]$.

Mutation Operators

Mutation operators which resemble biological mutation, alter one or a set of genes of a chromosome in pursuit of better ones. Two different mutation operators are used with different purposes. Mutation1 is used to yield feasible chromosomes or ones with lower degree of infeasibility. It switches assignment of one demand point to another facility if it ends to higher degree of feasibility. Mutation2 is used to improve a feasible solution. In a feasible solution, customers with same quantity of demands and different assignment are found and if swapping their facilities with each other leads to lower costs, their assignments will be swapped.

Stopping Criteria

These generation processes are repeated until reaching fixed number of generations. We choose 100 generations as stopping criteria at each iteration of subgradient optimization.

5. Computational Results

Test Problems

Three sets of different problems (10 problems per each set), each with different sizes and properties have been randomly generated to evaluate performance of proposed hybrid algorithm. The performance measure compares results of applying hybrid algorithm with optimal solution. Table 1 summarizes size, number of variables, service level of facilities and ratio between the total supply and the total demand for each set. This ratio depends on the demand distribution of customers and service level of facilities. Thus,
Tot supply/Tot demand that is reported in Table 1 is obtained after determining demands and service levels for each problem. Service levels are generated uniformly in the interval [0.8,1.0].

Table 1
The input parameters

Set	Problems	m	n	Number of variables	Service Level (α_i)	$^{\boldsymbol{Tot\ Supply}}/_{\boldsymbol{Tot\ Demand}}$
1	P1-P10	4	10	44	0.80-0.95	1.07-1.46
2	P11-P20	8	25	208	0.80-0.97	1.03-1.32
3	P21-P30	20	50	1020	0.80-0.98	1.00-1.46

The coordinates of facilities and demand points locations in set 1 are generated as U(0,100), U(0,200), U(0,300), U(0,400) and U(0,600). In set 2 coordinates are obtained from U(0,200), U(0,300), U(0,400) U(0,500) and U(0,1000) and in set 3 coordinates are generated as U(0,650), U(0,1000), U(0,1200), U(0,1500) and U(0,2000). (U(a,b) is uniform distribution in interval [a,b]). In the following, dispersion of candidate facilities and demand points in each problem is shown on 30 different scatter charts. The square markers in the charts represent location of candidate facilities, while the circle markers show the location of customers. Serving costs are generated as set 1 , part of set 3 and set 4 of test problems in Holmberg et al. (1999). They are determined as $c_{ij} = xd_{ij}$ where d_{ij} is Euclidean distance between facility i and customer j and x is a positive scalar. Customer demands generated from a uniform distribution in ranges [1,5] and [1,6]. We have tested various settings for fixed costs and amount of orders that each facility can serve. Fixed costs range from 100 to 1000 in set 1. In set 2 fixed costs range from 300 to 10000 and in set 3 from 1800 to 30000. Amount of orders served by facilities is generated as $U([1 \times y \times \frac{10}{4}], [5 \times y \times \frac{10}{4}])$ when customer demand ranges in $U(1,5)$ and as $U\left(\left[1 \times y \times \frac{10}{4}\right], [6 \times y \times \frac{10}{4}]\right)$ when customer demand is generated in $U(1,6)$. $[x]$ represents the greatest integer less or equal to x and y is a scalar equals to 1.1 , 1.2, 1.3 or 1.4.

The Tests

In Table 2, the results of computational tests are given. We have coded the hybrid algorithm in C language and all tests are implemented on a 2.40 GHz processor. For each of 30 problems, we provide both the optimal solution obtained with Optimization Toolbox of MATLAB and results of our hybrid algorithm as well as gap between these two values. The quality of proposed algorithm is validated through direct comparison with optimal solutions.

Table 2
The results of hybrid algorithm values versus optimal values

Problem	Optimal value	Hybrid algorithm value	Gap
P1	1319	1319	0.0000
P2	2213	2213	0.0000
P3	4143	4143	0.0000
P4	3592	3592	0.0000
P5	4506	4506	0.0000
P6	6184	6184	0.0000
P7	1238	1248	0.0080
P8	1620	1620	0.0000
P9	2805	2805	0.0000
P10	3787	3787	0.0000
P11	5055	5055	0.0000
P12	53256	53298	0.0008
P13	62885	63000	0.0018
P14	9749	9807	0.0059
P15	61963	62241	0.0045
P16	50341	50381	0.0008
P17	37031	37184	0.0041
P18	48133	48133	0.0000
P19	39937	39937	0.0000
P20	23736	23791	0.0023
P21	77292	77695	0.0052
P22	81501	82233	0.0090
P23	480088	481796	0.0035
P24	68132	68132	0.0000
P25	38445	38732	0.0075
P26	72977	73127	0.0021
P27	173891	174447	0.0032
P28	26408	26540	0.0050
P29	68255	68827	0.0084
P30	60076	60609	0.0087

The analysis of the computational results obtained by the hybrid algorithm, testifies the efficiency of proposed methodology for solving problems. The maximum gap, in all instances, between optimal objective function value and the value obtained by our hybrid algorithm is 0.9% and the average gap is 0.27%. In Table 3, maximum, minimum and average gap in different sets is presented.

Table 3
The summary of max/min and average gap

Set	Problem	Max Gap	Min Gap	Average Gap
1	P1-P10	0.0080	0.0000	0.0008
2	P11-P20	0.0059	0.0000	0.0020
3	P21-P30	0.0090	0.0000	0.0053

Table 4 indicates the number of iterations in hybrid algorithm that is repeated to achieve the solution. Moreover, number of branch and bound nodes or different cuts which is applied in order to gain optimal solution by the optimization toolbox of MATLAB is presented. As it is reported, while maximum number of iterations is 760, which is related to problem number 30, in problem 6, the hybrid algorithm was capable to obtain the optimal solution in just one iteration. In problem 22, optimization toolbox of MATLAB used 194627 branch and bound nodes to gain optimal solution, while 209 iterations of hybrid algorithm was adequate to yield a near optimal solution in less time with a gap equals to 0.0090.

Table 4
The results of hybrid algorithm iterations, branch and bound nodes, and number of cuts

		Optimal solution			
Problem	Hybrid Algorithm iterations	Branch and Bound Nodes	Cover cuts	Strong CG cuts	Gomory cut
1	11	8	-	-	-
2	518	-	-	1	-
3	2	-	4	3	-
4	8	-	3	3	-
5	12	-	1	1	-
6	1	-	2	4	-
7	7	4	-	-	-
8	17	-	1	1	-
9	5	-	2	4	-
10	24	-	10	3	1
11	342	14	-	-	-
12	464	62	-	-	-
13	255	70	-	-	-
14	219	17	-	-	-
15	16	94	-	-	-
16	56	12	-	-	-
17	44	-	11	2	-
18	5	-	9	5	-
19	51	-	5	3	1
20	113	-	6	5	-
21	190	9072	-	-	-
22	209	194627	-	-	-
23	252	2282	-	-	-
24	278	27494	-	-	-
25	589	208	-	-	-
26	418	109228	-	-	-
27	294	84511	-	-	-
28	181	1744	-	-	-
29	515	12929	-	-	-
30	760	4836	-	-	-

6. Conclusions

Stochastic capacitated facility location problem with single sourcing is formulated and a hybrid algorithm is presented to solve it efficiently. Set of potential capacitated facilities is to be selected to serve customers with uncertain demand while facilities fixed costs and costs of shipments are minimized. Each customer must be served only with one facility as the model is single sourcing. The power of proposed algorithm is embedded in its heuristic nature which is capable of solving SSSCFLP problems efficiently.

The hybrid algorithm consists of a Lagrangian heuristic coupled to an adjusted mixture of ant colony and genetic meta-heuristics. The Lagrangian relaxation is solved by solving a number of knapsack problems and the Lagrangian dual is solved by subgradient optimization. The computational results lead to the following conclusions: The Lagrangian relaxation together with subgradient optimization provide strong lower bounds to the problem. The adjusted mixture of ant colony and genetic optimization is very powerful in finding optimal or near optimal solutions.

The result of this work provides a perspective tool for modeling stochastic location-allocation decisions. This paper forms the basis of subsequent works which may consider routing, resource deployment, inventory holding and information sharing in the model or to be combined with simulation tools to compare different operational scenarios.

References

Neebe, A. W., & Rao, M. R. (1983). An algorithm for the fixed-charge assigning users to sources problem. *Journal of the Operational Research Society*, 34, 1107-1113.

Ahuja, R. K., Orlin, J. B., Pallottino, S., Scaparra, M. P., & Scutellà, M. G. (2004). A multi-exchange heuristic for the single-source capacitated facility location problem. *Management Science*, *50*(6), 749-760.

Barceló, J., & Casanovas, J. (1984). A heuristic Lagrangean algorithm for the capacitated plant location problem. *European Journal of Operational Research*,*15*(2), 212-226.

Beasley, J. E. (1993). Lagrangean heuristics for location problems. *European Journal of Operational Research*, *65*(3), 383-399.

Chen, C. H., & Ting, C. J. (2008). Combining lagrangian heuristic and ant colony system to solve the single source capacitated facility location problem.*Transportation research part E: logistics and transportation review*, *44*(6), 1099-1122.

Cortinhal, M. J., & Captivo, M. E. (2003). Upper and lower bounds for the single source capacitated location problem. *European journal of operational research*,*151*(2), 333-351.

Holmberg, K., Rönnqvist, M., & Yuan, D. (1999). An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, *113*(3), 544-559.

Hindi, K. S., & Pieńkosz, K. (1999). Efficient solution of large scale, single-source, capacitated plant location problems. *Journal of the operational Research Society*, *50*(3), 268-274.

Tragantalerngsak, S., Holt, J., & Rönnqvist, M. (2000). An exact method for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, *123*(3), 473-489.

Klincewicz, J. G., & Luss, H. (1986). A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints. *Journal of the Operational Research Society*, *37*(5), 495-500.

Lin, C. K. Y. (2009). Stochastic single-source capacitated facility location model with service level requirements. *International Journal of Production Economics*, *117*(2), 439-451.