Journal of Uncertain Systems
Vol.12, No.2, pp.123-140, 2018
Online at: www.jus.org.uk

A Bi-objective Multi-facility Location-Allocation Problem with Probabilistic Customer Locations and Arrivals: Two Meta-heuristics Using Discrete Approximation

Roshanak Mohammadivojdan[1], Seyed Taghi Akhavan Niaki[2,*], Mohsen Dadashi[3]
[1]*Department of Industrial & Systems Engineering, University of Florida*
[2]*Department of Industrial Engineering, Sharif University of Technology,*
P.O. Box 11155-9414 Azaadi Ave., Tehran 1458889694 Iran
[3]*Department of Industrial Engineering, University of Arkansas*

Received 9 October 2017; Revised 31 December 2017

Abstract

In this work, a bi-objective multi-facility location-allocation problem is investigated, in which the locations of the customers and their arrivals are stochastic. We first formulate the problem as a continuous location-allocation model with no constraints on the capacity of the facilities. Then, we develop an approximated discrete model in which the facilities with limited capacities can be located on a set of candidate points. The proposed model has two objective functions that are evaluated using discrete event system simulation. The first objective is to minimize the expected total time the customers spend in the system until their services begin. The time that each customer spends in the system includes the customer's travel time as well as his/her waiting time in the facility until he/she receives service. The second objective is to minimize the sum of the expected queue lengths. Considering the NP-hardness of the problem and the unique properties of the objective functions, a Non-dominated Sorting Genetic Algorithm (NSGA-II) is developed to obtain a Pareto optimal front. We have proposed a heuristic approach for generating feasible solutions to initiate NSGA-II. Since there is no benchmark available in the literature, in order to evaluate the obtained results, we have utilized another multi-objective meta-heuristic approach called Non-dominated Ranked Genetic Algorithm (NRGA). For further validation, we have also employed a genetic algorithm to solve two single-objective problems separately.
© 2018 World Academic Press, UK. All rights reserved.

Keywords: multi-facility location-allocation, genetic algorithm, NSGA-II

1 Introduction

The multi-facility location-allocation problem is concerned with locating n facilities in the Euclidian plane and allocating m customers to those facilities while minimizing the total cost. This cost mostly consists of the transportation cost. Suppliers, warehouses, producers, or servers can be counted as facilities, and retailers, purchasers, and service users are usually treated as the customers. Cooper [9] was the first who introduced the location-allocation problem (LAP). This problem has been studied in various settings; if the location of a facility can be selected from a set of candidate points, the problem is a discrete optimization problem. In some other settings, however, the facilities might be placed at any point in the Euclidian plane, and the problem is continuous and is known as the multi-facility Weber location problem (MFWP) in the literature [37]. The Weber problem which was first introduced by Weber [35], is to locate a single facility in the plane, where the weighted sum of its distances from the customers is minimized. While various MFWPs have been proposed in the literature, the most recent is studied by Jiang et al. [20] who proposed a generalized variation of the Weber problem on a plane in which a straight line divides the plane into two regions and various gauges are used to measure the distances in different regions.

Sherali and Nordai [33] showed that an MFWP with capacity limitations on facilities and deterministic parameters is NP-hard even if all customers are located on a straight line. Consequently, the exact methods are not

[*] Corresponding author.
Emails: r.mohammadivojdan@gmail.com (R. Mohammadivojdan), Niaki@sharif.edu (S. T. A. Niaki), mdadashi@uark.edu (M. Dadashi).

practical in solving those settings. Although there are certain conditions under which the problem is solvable. For large-scale problems the only reasonable approach is to use heuristic or meta-heuristic methods.

Cooper [10] introduced the first heuristic approach for solving MFWP. This well-known heuristic is based on the fact that the two components of the problem, namely finding the locations of the facilities and allocating customers to them, are easier to solve separately. This algorithm which alternates between location and allocation sub-problems is called ALA and has been used as a measure for other algorithms since then. Cooper's ALA algorithm has also been playing a part such as improving the initial solutions in some solution algorithms. Love and Juel [25] introduced the neighborhood search method for the first time and then Mladenovic and Brimberg [26] combined their idea with the Cooper's ALA method to propose a descent-ascent-based method. They also used Tabu search and variable neighborhood algorithms to solve an un-capacitated continuous location-allocation problem [6, 7]. Later, Salhi and Gamal [32] proposed an improved genetic algorithm (GA) to find near-optimal solutions for this problem. Taillard [34] proposed a heuristic method based on decomposition, and Brimberg et al. [4] extended his work by developing a variable neighborhood decomposition search (VNDS) heuristic that was applicable to variable sizes of the subproblems. Hosseininezhad et al. [17] developed a crossed-entropy-based meta-heuristic for the capacitated multi-source Weber problem considering a fixed cost for opening each facility. They presented a mixed integer nonlinear formulation and then utilized a three-stage cross entropy meta-heuristic to solve the problem. They showed that for small-sizes of the problem, their approach performs well compared with commercial solvers. Recently, Drezner and Salhi [13] proposed two methods to reduce the neighborhood for the solution of the p-median problem on a plane. Brimberg et al. [5] provided a survey on solution methods for the continuous location-allocation problem.

Various multi-objective optimization algorithms have been proposed in the literature to solve facility location-allocation problems. Farahani et al. [15] provided a good review and classification of such algorithms. Pasandideh et al. [30] developed a multi-objective facility-location problem within the batch arrival queuing framework. Their objectives include (1) minimizing the weighted sum of the waiting and traveling times, (2) minimizing the maximum idle time of the facilities, and (3) minimizing the total facility establishment cost. They implemented a simulated annealing (SA) and a genetic algorithm (GA) to solve their problem. In a recent work, Hajipour et al. [16] discussed a multi-objective multi-layer facility location-allocation model for congested facilities. The objectives were (1) minimizing the sum of travel and waiting times; (2) minimizing the establishment cost of the facilities; and (3) minimizing the maximum idle probability of the facilities. They utilized a multi-objective vibration damping optimization (MOVDO) as well as a multi-objective harmony search (MOHS) algorithm to find Pareto solutions.

Although early efforts were devoted to modeling and solving LA problems with the assumption of the parameters being deterministic, in many real-world applications this assumption is not reasonable. An example involves locating emergency stations for demands with uncertain locations. A very common approach to this type of problems is to assume that the uncertain parameters are probabilistically distributed. Katz and Cooper [21] studied a single facility LA problem with probabilistic customer locations. In their work, locations of the customers follows a symmetric bivariate normal distribution, where the Euclidian function was used to calculate the distances. Later, Katz and Cooper [22] considered the case of symmetric bivariate exponential distribution for customer locations as well. Wesolowsky [36] investigated the case of the rectilinear distance function. Logendran and Terrell [24] modeled an un-capacitated continuous LA problem with uncertain demands. Carrizosa et al. [8] investigated and solved a more general case where the locations of the facilities and customers, both could be stochastic. Moreover, Zhou [38] utilized the expected value model (EVM) and chance-constrained programming (CCP) for uncertain demands of an un-capacitated continuous LA problem. In another research, Zhou and Liu [39] introduced the dependent-chance programming (DCP) for un-capacitated as well as capacitated continuous LA problems with stochastic demands. They combined the network simplex algorithm, stochastic simulation, and genetic algorithm to solve the problem through a hybrid intelligent algorithm. Jamalian and Salahi [19] developed an equivalent formulation for the robust counterpart of MFWP with uncertain locations for the demand points and transportation cost.

Aras et al. [2] studied a capacitated continuous location-allocation problem in both deterministic and stochastic settings. For the deterministic version of the problem, they proposed a discrete approximation model and a heuristic solution algorithm. For the stochastic case, where locations of the customers were probabilistically distributed, they proposed a mathematical model and a heuristic solution algorithm based on Cooper's [10] ALA algorithm. In another work, Aras et al. [3] extended this algorithm and proposed two other algorithms using a discrete approximation. Pasandideh et al. [30] investigated a multi-objective facility location problem with batch arrivals and developed two parameter-tuned genetic and simulated annealing algorithms to solve the problem. Mousavi et al. [27] proposed a new model for the capacitated location-allocation problem where both locations and demands of the customers were stochastic. Moreover, Rahmati et al. [31] formulated a tri-objective facility location problem within multi-server queuing framework, in which selecting the nearest-facility along with the service level restriction were considered to bring the model closer to reality. They proposed a Pareto-based meta-heuristic to solve the problem.

In this paper, we consider a setting for the continuous multi-facility LA problem, with a specially structured stochastic demand. We assume the location of each customer can be modeled as a two-dimensional random vector that follows a bivariate normal distribution. This could refer to a situation in which we are trying to locate a set of service centers to provide a service for customers using certain types of products (vehicles for instance). This could be even more valuable when the distance of the demand location with the service station is more sensitive. In this particular setting, it is reasonable to allocate the customers to the stations (provide them the information of where they can get help with their product) instead of having the policy of going to the nearest station. Although the stochastic locations and arrival times on the side of customers has been studied in the literature previously, previous studies either model the location of the next customer as a Poisson random variable on the two dimensional plane, or in case customers are individuals with independent location random variables, they assume each customer goes to the nearest service center. In this work, however, we consider a continuous multi-facility LA problem with stochastic customer locations and arrivals where the customers assigned to a certain facility will always be served in that facility. To the best of our knowledge, this particular problem setting has not been studied in the literature yet.

We first provide a mathematical formulation of this problem. Then, we discuss an approximated discrete model with two objectives in which facilities with limited capacity can be located on a set of candidate points (a case that is closer to reality with many applications). Since the problem is NP-hard, we utilize a multi-objective evolutionary algorithm, the nondominated sorting genetic algorithm (NSGA-II) to find Pareto optimal front, and we propose a new heuristic approach to generate initial feasible solutions. Further, as no benchmark is available in the literature, another meta-heuristic multi-objective evolutionary algorithm, the non-dominated ranked genetic algorithm (NRGA) is also employed to validate the results obtained. For further validation, the results are also compared with the ones obtained employing a genetic algorithm to solve two single-objective problems separately. The solution algorithms are then evaluated using some multi-objective performance measures.

The remainder of the paper is organized as follows. In Section 2, the un-capacitated continuous location-allocation problem with stochastic customer locations and arrivals is introduced, and the mathematical formulation of the problem is presented. Then, an approximated discrete model is proposed in this section to make the problem easier to solve. In Section 3, not only a genetic algorithm is developed for the single-objective version of the problem, but also NSGA-II and NRGA are utilized to solve the multi-objective version of the problem. The computational results are discussed in Section 4 and finally, Section 5 concludes.

2 Problem Description and Formulation

This section describes the problem, the assumptions made, and the proposed formulation. The goal of the problem is to locate a set of un-capacitated facilities with specific service rates on the Euclidian plane and to allocate a set of customers to these facilities. The model that will be developed in this paper accounts for a set of assumptions and conditions, along with a specially structured stochastic demand. We assume that the location of each customer can be modeled as a two-dimensional random vector that follows a bivariate normal distribution with known parameters. Further, the demand for each customer is assumed to follow a Poisson process that could fairly describe many demand processes. Using the same argument, we also assume that the service times of the customers are independent exponential random variables.

The above approach for modeling the demand locations could pertain to a variety of practical situations and has been used previously in the literature. For example, when the customers' addresses and the average distances they travel every day are known, one could derive the parameters of the normal distribution for each customer. Although the stochastic locations and arrival times on the side of customers has been studied in the literature, previous studies either model the location of the next customer as a Poisson random variable or they make the assumption that each customer goes to the nearest service center. In the current work, however, we account for stochastic customer locations and arrivals, with focus on a situation where customers are allocated to a certain facility, and seek service in that facility.

The travel time of each customer to a specific facility depends on its location as well as its travel speed; we assume a constant speed for all customers. Nevertheless, this assumption can be relaxed if one has the data for the existing paths and the travel times associated with each path, which is a possible direction for future works on this problem. Hence, at each particular facility, the entrance rate of the customers is affected by the demand occurrence rate of the customers assigned to that facility and their travel times, both of which being stochastic. Consequently, one may assume that each facility behaves as a G/M/1 queue.

2.1. Problem Formulation

In order to formulate the continuous LA problem discussed above, the following notations are used:

Indices:

i : Index for facilities (i : $1, \dots, n$)
j : Index for a customers (j: $1, \dots, m$)

Parameters:

$a_j = (a_{j1}, a_{j2})$: Stochastic location point of customer j
λ_j : Demand rate of customer j
μ : The common service rate of facilities
c : The common travel speed of customers ($distance\ unit/time\ unit$)

Decision variables:

$X_i = (X_{i1}, X_{i2})$: Location of facility i
$Y_{ij} = \begin{cases} 1 & if\ customer\ j\ is\ allocated\ to\ facility\ i \\ 0 & otherwise \end{cases}$

The total demand rate at each facility is equal to the sum of the demand rates of the customers allocated to that facility. In other words,

$$The\ expected\ total\ demand\ rate\ at\ facility\ i = \gamma_i = \sum_{j=1}^{m} \lambda_j Y_{ij}. \tag{1}$$

The first objective function to be minimized, represented by Z_1, is the expected total time the customers spend until their services begin. This customer-based objective includes the sum of the expected travelling times and the expected waiting times in the queue. The reason why this objective function has been chosen can be explained using an example. Consider a situation in which one is trying to find the best location among a set of ATMs (automated teller machines) based on the time she/he travels in addition to the time she/he spends in the waiting line. As the queue reaches equilibrium in the long term, the average waiting time can be considered as a decision criterion. Although one might think of an imaginary allocation step for this particular example, to make sure the best realization of customers' allocation is consistent with the best locations found and in that case, since customers do have a choice, this performance for a queue in its equilibrium state may not be optimal, but the solution found may be a suggestion for the customers.

The first part of this objective function is the sum of the travel times of the demanding customers and the second part is the expected sum of their waiting times [29], i.e.

$$Z_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j \times \frac{E\left(d(X_i, a_j)\right)}{c} \times Y_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j \times w_i \times Y_{ij}\ . \tag{2}$$

In Eq. (2), $E\left(d(X_i, a_j)\right)$ is the expected Euclidean distance between the locations of facility i and customer j obtained based on a procedure explained in Section 4, c is the travel speed (hence $E\left(d(X_i, a_j)\right)/c$ is the expected travel time), and w_i is the expected waiting time of a customer at facility i. In order to calculate w_i, we need to know the distribution of the demand entrance to each facility, which cannot be found analytically. Consequently, we have utilized a simulation technique to estimate values of w_i in each scenario. As explained earlier, the time between two arrivals to a certain facility consists of two components: time between occurrence of two demands and the travel time of customers to the facility. Hence, the inter-arrival time to a facility is the sum of an exponentially distributed random variable and a bivariate normal distribution for the travel time.

The second objective is to minimize is the expected sum of the queue lengths represented by Z_2. This system-based objective can also be explained through the ATM example, where this time the factor affecting the decision would be the average number of waiting customers in the queue. That is

$$Z_2 = \sum_{i=1}^{n} L_i\ . \tag{3}$$

In Eq. (3), L_i represents the expected queue length at facility i, which is another parameter of the G/M/1 queue.

As a result, the mathematical model for the uncapacitated multi-facility continuous LA problem with stochastic customer locations and arrivals is

$$Min\ \ Z_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j \times \frac{E\left(d(X_i, a_j)\right)}{c} \times Y_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j \times w_i \times Y_{ij}$$

$$Min\ \ Z_2 = \sum_{i=1}^{n} L_i$$

$s.t.$

$$I)\ \sum_{i=1}^{n} Y_{ij} = 1 \qquad \forall\, j : j = 1, \dots, m \tag{4}$$

$$II)\ \sum_{j=1}^{m} \lambda_j\, Y_{ij} \le \mu \qquad \forall\, i : i = 1, \dots, n \tag{5}$$

$$Y_{ij} \in \{0,1\} \qquad i = 1, \dots, n\ \ ,\ \ j = 1, \dots, m.$$

In this model, the first constraint guarantees that each customer is allocated to exactly one facility and the second constraint is set to check the stability of each queue, where it means that the demand rate of each facility should be less than its service rate.

2.2. Discrete Approximation Model

Aras et al. [2, 3] proposed a mixed integer linear programming (MILP) model based on an approximation using vertices of a grid covering all customer locations. In order to illustrate the idea better, the example that was presented in Aras et al. [2] is used here. Consider the location problem shown in Figure 1. In this figure, the square marks represent customer locations. It is known that there is a set of optimal facility locations that lie in the convex hull made by the set of customer locations. Considering the smallest rectangle that covers all of these customer locations, we divide it using horizontal and vertical lines of the same distance from one another. The resulting grid points within the convex hull are the candidate points for the locations of the facilities.

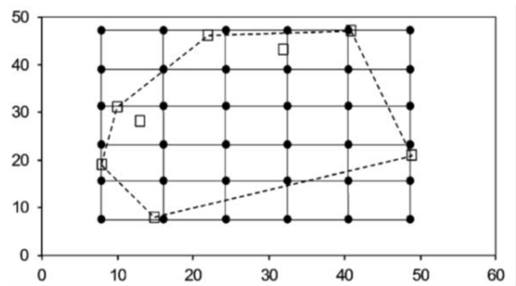

Figure 1: Candidate points of a small size problem (Aras et al. 2007)

The idea of using candidate points in order to propose a discrete approximation for the LA problem can be used for the stochastic version of the problem as well. Clearly, as the number of candidate points increases, the resulting solution will come closer to the optimal solution. If the number of points goes to infinity, or equivalently the distance between dividing lines goes to zero, the solution to the approximated problem will be an optimal solution to the continuous version.

In our problem, if t candidate points are in the convex hull built by customer locations, the coordinates of the k-th candidate point ($k = 1,2, \dots, t$) will be represented by $b_k = (b_{k1}, b_{k2})$ and the decision variables will change to:

$$U_k = \begin{cases} 1, & if\ a\ facility\ is\ located\ at\ candidate\ point\ k \\ 0\,, & otherwise, \end{cases}$$

$$Y_{kj} = \begin{cases} 1, & if\ customer\ j\ is\ allocated\ to\ facility\ k \\ 0, & otherwise. \end{cases}$$

Then, the first part of the first objective function representing the sum of all travel times will be:

$$\textstyle\sum_{j=1}^{m} \sum_{k=1}^{t} \lambda_j \times \frac{E\left(d(b_k, a_j)\right)}{c} \times U_k \times Y_{kj}\ . \tag{6}$$

And the sum of waiting times is:

$$\sum_{j=1}^{m}\sum_{k=1}^{t} \lambda_j \times w_k \times Y_{kj}\ \ . \tag{7}$$

The second objective function will also change as follows:

$$Z_2 = \sum_{k=1}^{t} L_k \tag{8}$$

Thus, the discrete approximation of the model can be written as:

$$Min \quad Z_1 = \sum_{j=1}^{m}\sum_{k=1}^{t} \lambda_j \times \frac{E\left(d\left(b_k, a_j\right)\right)}{c} \times U_k \times Y_{kj} + \sum_{j=1}^{m}\sum_{k=1}^{t} \lambda_j \times w_k \times Y_{kj}$$

$$Min \quad Z_2 = \sum_{k=1}^{t} L_k$$

s.t.

$$I)\ \sum_{k=1}^{t} Y_{kj} = 1 \qquad \forall\, j : j = 1, \dots, m \tag{9}$$

$$II)\ Y_{kj} \le U_k \qquad \forall\, j : j = 1, \dots, m \ ,\ \ \forall\, k : k = 1, \dots, t \tag{10}$$

$$III)\ \sum_{j=1}^{m} \lambda_j\, Y_{kj} \le \mu \qquad \forall\, k : k = 1, \dots, t \tag{11}$$

$$IV)\ \sum_{k=1}^{t} U_k = n \tag{12}$$

$$Y_{kj} \in \{0,1\}$$

$$U_k \in \{0,1\}.$$

In this model, the first constraint guarantees that each customer is allocated to exactly one facility. The second constraint will make sure that each customer can only be assigned to a candidate points where facilities are located. The third constraint is set to check the stability of each queue, and finally, the fourth constraint is set to make sure that all facilities (no more than the number of available facilities) are located.

The optimal solution of the above-approximated model is not necessarily optimal to the original problem, but an increase in the number of candidate points will reduce the gap. However, by increasing the number of candidates the computational effort needed to solve the problem will increase as well. Consequently, considering the desired accuracy, one should make a balance between the quality of the solution found for the continuous model and the computational effort needed to solve the discrete model.

3 Solution Methods

As mentioned in the previous section, calculation of the objective functions requires us to compute parameters of the G/M/1 queue at each facility. Since the distribution of demand arrivals to each facility is unknown, we utilize a simulation approach to estimate these parameters. This complicates the problem and makes it hard to solve optimally (even for small sizes of the problem) as opposed to many other LA problems that can be solved through exact solution approaches. This leaves us with no other choice except utilizing heuristic or meta-heuristic methods.

Considering the special characteristics of the problem, a search-based algorithm is required. Some algorithms such as simulated annealing, Tabu search, and genetic algorithm reduce the probability of being trapped in local optimum solutions. Among them, genetic algorithm (GA) has performed fairly well in LA models with similar structure (see for example [5, 32]). Although many heuristics or meta-heuristics have the property of producing several solutions at a given stage, they usually use a neighborhood search method. Consequently, their solutions do not have the extensity and diversity of GA solutions unless diversification methods are used.

Note that the two objective functions introduced for this problem cannot be merged easily due to the difference in their entities. Moreover, as the first objective function includes the travel times to the facilities, it can conflict the second objective function i.e. the average queue lengths. This leads to taking advantage of multi-objective programming approaches to solving this problem. One common approach is to search for a set of Pareto-optimal solutions. In a multi-objective problem with l objective functions, if all objectives are in the minimization form, the feasible solution y is dominated by the feasible solution x if

$$x > y \ (x \ dominates \ y) \ \leftrightarrow \begin{cases} \forall\, i\ (i{:}\,1, \dots, l){:}\ \ z_i(x) \le z_i(y) \\ \qquad\qquad and \\ \exists\, i : \qquad\qquad z_i(x) < z_i(y). \end{cases} \tag{13}$$

A feasible solution is said to be Pareto-optimal if it is dominated by no other solution in the feasible space. The set of all Pareto-optimal solutions in the feasible space is called Pareto-optimal set and the values of their objective functions are called Pareto-front. Among several multi-objective evolutionary algorithms proposed recently, NSGA-II is one of the best and most efficient algorithms. This algorithm and another similar algorithm called NRGA are chosen to solve the problem.

In the rest of this section, we first describe how a genetic algorithm is implemented to solve the problem considering each of the objectives in isolation. Then, we explain the solution approaches for solving our bi-objective problem using NSGA-II and NRGA. A heuristic method to produce initial solutions will also be proposed.

3.1. Genetic Algorithm

In the genetic algorithm (GA), an individual or a chromosome presents a solution. Chromosomes consist of units called genes. Each gene defines a characteristic of a chromosome. GA operates on a set of chromosomes called population. The first population is usually produced randomly. During the successive iterations (generations), only the best chromosomes with respect to a quality measure, usually a fitness function, remain in the population and finally, the algorithm converges to the best chromosome (solution).

GA uses two operators to produce new chromosomes: crossover and mutation. The crossover operator, which is the more important operator, combines two chromosomes called parents to produce new chromosomes called children. The mutation operator makes random alterations in chromosomes. In order to select parents, a selection operator is used. In single objective problems, this selection is based on a fitness function, and the fitness function is often the objective function. In multi-objective problems, however, the selection operator acts differently.

The chromosome we designed for the GA algorithm consists of two main parts, **U** and **Y** shown in Figure 2, where $\mathbf{U}$ is a $t \times 1$ vector representing the first decision variable. The elements of this chromosome indicate the candidate points 1 to t where the facilities are to be located. In other words, **U** takes the form $[Z_1, \dots, Z_k, \dots, Z_t]^T$. The second part, **Y** is a matrix that shows the allocation of customers to facilities. Each column of this matrix refers to a specific customer and it takes value 1 if the customer is allocated to a facility, otherwise zero. The matrix **Y** takes the form

$$\begin{bmatrix} Y_{11} & \dots & Y_{1m} \\ \vdots & Y_{kj} & \vdots \\ Y_{t1} & \dots & Y_{tm} \end{bmatrix}_{t\times m}.$$

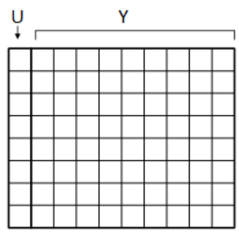

Figure 2: Chromosome representation

3.1.1. The Crossover Operator

This operator combines two parents in order to produce two children. It combines the **Z** vector of one parent with the **Y** matrix of another to produce a child, as shown in Figure 3.

3.1.2. The Mutation Operator

Two mutation operators are defined in GA as:
Mutation operator 1: It chooses one of the non-zero elements of **Z** randomly and moves it with its customers to another row with a zero value. In other words, it chooses a facility randomly and moves it with all of its customers.
Mutation operator 2: This operation randomly selects a customer and allocates it to another facility.

Examples of these operations are shown in Figure 4.

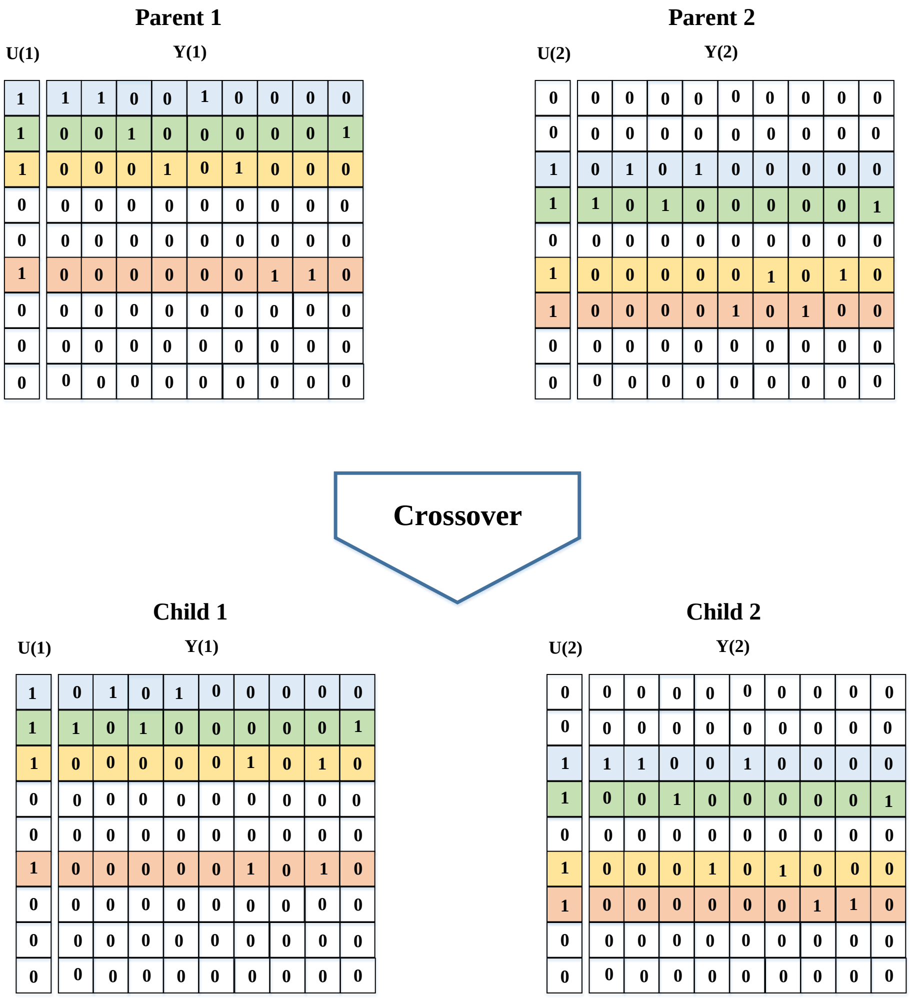

Figure 3: Crossover operation

3.1.3. The Selection Operator

This operator helps to select parents in order to produce next generation. For a single-objective model, the value of the objective function is considered as the fitness function, where a chromosome with a lower fitness function value is better and more probable to be chosen.

3.1.4. Feasibility of a Solution

Feasibility of a solution found by this algorithm should be guaranteed somewhere in the process. There are different approaches to check feasibility. When the number of constraints is large, one efficient way to check feasibility is to use a penalty function. Since the number of constraints is small in our developed model, we use the following rules to assure feasibility [11]:

1) A feasible solution is always preferred to an infeasible solution.
2) Between two feasible solutions, the one with better objective function value is preferred.
3) Between two infeasible solutions, the one with smaller constraint violation is preferred.

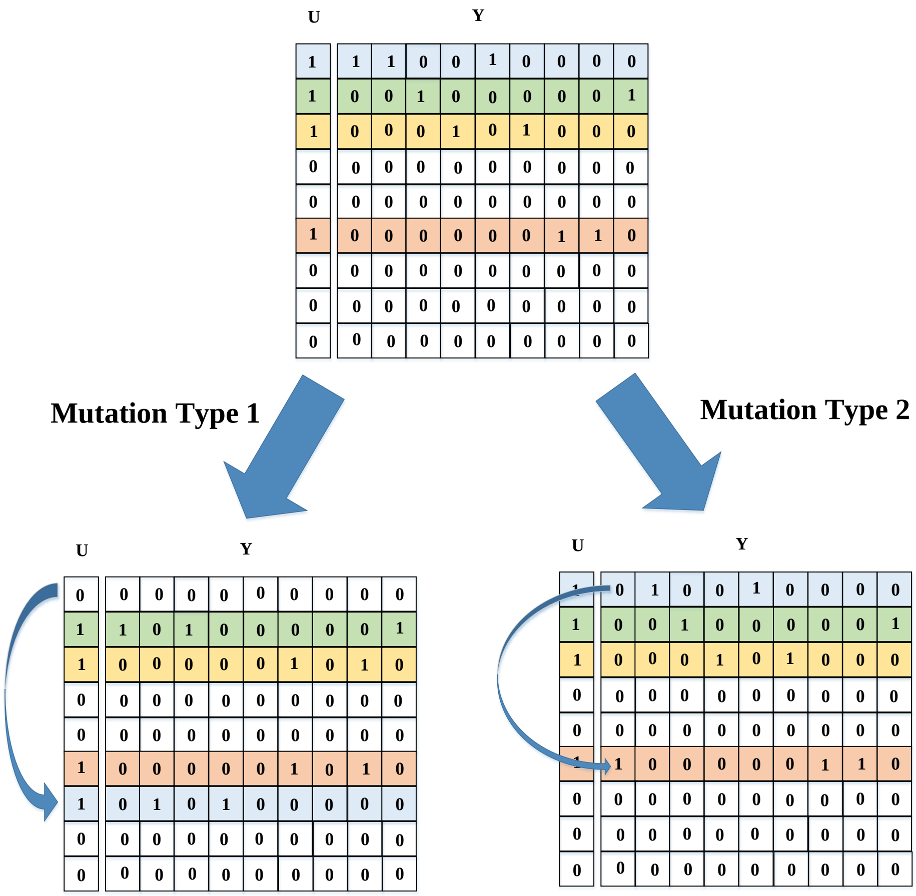

Figure 4: Mutation operation

3.2. The NSGA-II

One of the most popular algorithms for solving multi-objective optimization problems is the so-called elitist non-dominated sorting genetic algorithm (NSGA). It is a simple and effective algorithm that requires minimum user interaction. An improved extension of NSGA is named NSGA-II that has been utilized in many optimization problems, efficaciously [12, 23]. NSGA-II, with a better sorting property compared to NSGA in the sense that it incorporates elitism and does not require any sharing parameter to be chosen in advance, takes advantage of the dominance concept to select solutions for the Pareto front. There are five operations namely initialization, fast non-dominated sorting, crossover, mutation, and the elitist crowded-comparison operation, involved in the original version of NSGA-II [28]. We used these operations in the NSGA-II that we developed for the purpose of this study.

The crossover and the mutation operators are the same operators defined in the GA described in Section 3.1. However, since there are more than one objective functions, it is not possible to sort chromosomes with respect to their fitness functions. Instead, the following two criteria are used in order to sort and select a better chromosome:

1) Non-domination level,

2) Crowding distance.

In order to sort the chromosomes, the fast non-dominated sorting approach is first employed to divide them into different levels of nondomination. Then, the crowding distance is used to sort each level [12]. After sorting the population, the selection operator chooses a set of parents randomly. This selection is based on a probability distribution such that better solutions are more likely to be chosen, where

$$(i \text{ is better than } j) \quad if \begin{cases} (i_{rank} < j_{rank}) \\ or \\ (i_{rank} = j_{rank}) \text{ and } (i_{distance} > j_{distance}). \end{cases} \tag{14}$$

In Equation (14), i_{rank} refers to the non-domination level and $i_{distance}$ refers to the crowding distance value for solution i. Moreover, the feasibility of a solution is guaranteed using the same rules as the ones used in the GA with the following changes in the second rule: between two feasible solutions, the solution with lower non-domination level is preferred; if they belong to the same level of non-domination, the solution having greater crowding distance value is preferred. The main steps involved in NSGA-II are described in the following subsection.

The main steps of the NSGA-II algorithm are

1. An initial population (P_0) of the chromosomes is created. Although it can be produced randomly, in order to produce feasible solutions and improve the quality of the initial population, a heuristic method described in Section 3.4 is used in this step.
2. The population P_0 is sorted with respect to the non-domination level and the crowding distance assignment.
3. Parents are selected using the selection operator.
4. Crossover and mutation operators are applied to produce children population.
5. The population of children is first combined with the previous generation. Then, the new generation is selected from the sorted combined population.
6. A predefined number of randomly produced chromosomes are added to the new generation.
7. Repeat (3)-(6) until the stopping criterion is satisfied.

3.3. The NRGA

This algorithm is similar to NSGA-II. The only difference is that the selection operator could end up with better results in some cases [18]. NRGA utilizes a ranked based roulette wheel selection. In this algorithm, each solution is assigned a rank based on its non-domination level and the probability of selecting an individual which is defined in Equation (15).

$$Pi = \frac{2 \times (max - rank)}{N(N+1)} \tag{15}$$

In Equation (15), max is the maximum rank available in the population, $rank$ refers to the chromosome's non-domination level, and N is the number of individuals in the population. A chromosome with a lower rank is more likely to be selected. In our problem, the chromosomes of the population are first arranged according to their non-domination level and a level is chosen using ranked based roulette wheel selection. Then, chromosomes of the same rank are sorted by the crowding distance measure and again a chromosome is chosen among them.

3.4. The Proposed Heuristic Approach to Generate Initial Solutions

This heuristic is proposed in order to make feasible and preferable initial solutions. It first chooses random locations for facilities using candidate points. Then, the distances between each customer and all facilities are calculated. Next, starting from the nearest facility, the allocation process begins. If the allocation does not disturb the stability of the queue for the corresponding facility, the customer will be allocated to that facility. Otherwise, the next nearest facility is considered. This process continues until all customers are allocated. The flowchart of this heuristic is shown in Figure 5.

3.5. The Simulation Approach

As described in the previous sections, in order to calculate the objective functions we use a discrete system simulation approach to estimate the required parameters of the G/M/1 queues. The simulation works as a part of the coded algorithm (GA, NSGA-II or NRGA that are coded in C#). Each solution of the problem indicates where the facilities are located and how the customers are allocated to them. Considering each solution, there are n systems of G/M/1 queues that should be simulated. In order to simulate this system, the following assumptions are made:

- There is no limit on the number of customers waiting in each queue.
- The time between the occurrence of a demand and its arrival to the related facility is treated as the travel time.
- The simulation starts considering zero values for all variables such as queue lengths and the number of busy facilities.

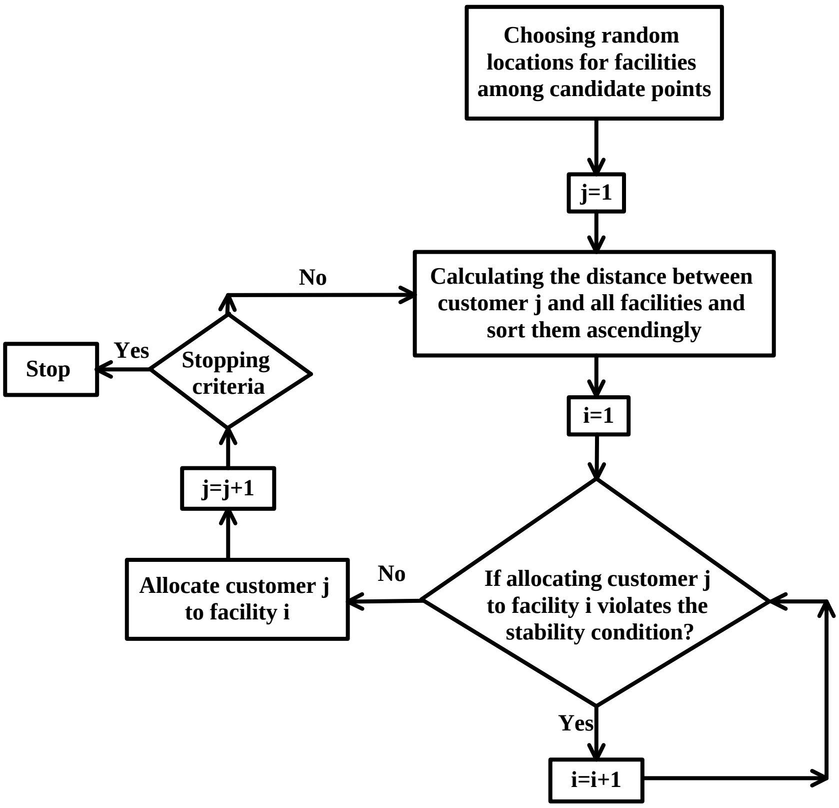

Figure 5: Flowchart of the proposed heuristic

The simulation starts at time zero with no customers in line. Then, we start to generate the following random attributes for the next demand of each customer:

1. Time until the next event
2. Coordinates of the location for the next demand
3. The service time associated with this demand

Attribute 1 is generated randomly based on an exponential distribution (because the demand is assumed to follow a Poisson process) with parameters specific to the customer. Attribute 2 will be generated randomly based on the bivariate normal distribution for that customer's demand location. Attribute 3 is also generated according to an exponential distribution.

At each step, one event is planned to occur, where the first event is the customer with the smallest value for Attribute 1. Then, the travel time is calculated based on the customer's distance to his/her associated facility, based on which the customer is assigned to that line. These steps are followed for the time interval specified for the simulation. One can calculate the time each customer spends in the queue and also the number of people in line.

The simulation approach provides the following parameters:

- The expected waiting time for a customer at facility i

- The expected queue length at facility i

Consequently, calculation of the objective functions for each feasible solution is possible through the expected waiting time and the expected queue length given in Equations (16) and (17).

$$\begin{aligned} &\mathit{The\ expected\ waiting\ time\ of\ a\ customer\ at\ a\ facility} \\ &= \frac{\mathit{Total\ time\ spent\ by\ customers\ in\ the\ facility's\ queue}}{\mathit{Total\ number\ of\ customers\ entered\ the\ facility}} \end{aligned} \tag{16}$$

$$\begin{aligned} &\mathit{The\ expected\ queue\ lenght\ at\ a\ facility} \\ &= \frac{\mathit{sum\ of}(\mathit{queue\ length}\ \times \mathit{the\ time\ during\ which\ the\ queue\ length\ is\ constant})}{\mathit{simulation\ time}} \end{aligned} \tag{17}$$

4 Computational Results

In this section, we present and evaluate the results obtained by the developed algorithms. First, we discuss the method used to calculate the expected distances. Then, several numerical problem instances are examined.

4.1. Evaluating Expected Distances

In order to compute the first part of the first objective function, the expected distances between customers and facilities are required. For a symmetric bivariate normal distribution, the expected distance between the facility i and customer j, $E\left(d(X_i, a_j)\right)$, is obtained by

$$E[d(X_i, a_j)] = \int_{-\infty} \int_{-\infty} d(X_i, a_j)\, f_j\,(a_j) da_{j1} da_{j2}. \tag{18}$$

In Equation (18), a_{j1} and a_{j2} are the coordinate components, which are independently distributed (i. e., $_{j1j2} = 0$), with expected values μ_{j1} and μ_{j2} and standard deviations $\sigma_{j1} = \sigma_{j2} = \sigma_j (j = 1, \dots n)$. The integration in (18) is not easy to calculate, especially when the distribution of the distances is not known. In this work, the approximation method proposed by Altınel et al. [1] and Durmaz et al. [14] is used. The approximation function is

$$g(x) = \begin{cases} d(x,\mu) + \dfrac{1}{2}\dfrac{\sigma^2}{d(x,\mu)} & if \quad d(x,\mu) \geq \dfrac{\sigma}{\sqrt{2}} \\ \sqrt{2}\sigma & if \quad d(x,\mu) < \dfrac{\sigma}{\sqrt{2}}. \end{cases} \tag{19}$$

In Equation (19), $d(x, \mu)$ refers to the Euclidian distance between x and μ.

4.2. Results

In order to calculate the objective functions for each feasible solution, simulation of the entire system is required. This makes it impossible to find the exact optimal solution or the Pareto-Front in the multiobjective case. Consequently, we chose to solve two single-objective problems separately through the genetic algorithm in order to evaluate the results obtained by NSGA-II and NRGA.

Several problem instances are presented in this section. We first discuss a problem and its results step by step. The problem presented here has 10 customers and 5 similar facilities to locate. As mentioned earlier, each customer location follows a bivariate normal distribution. The location of customer j is $a_j = (a_{j1}, a_{j2})$ with expected value $\mu_j = (\mu_{j1}, \mu_{j2})$ and standard deviation $\sigma_j = (\sigma_{j1}, \sigma_{j2})$ where $\sigma_{j1} = \sigma_{j2} = \sigma_j$ (the coordinates are independantly distributed with a unique standard deviation). In this example, both μ_{j1} and μ_{j2} are generated using a uniform distribution in interval [0,1000] and the variances are also created using a uniform distribution in [0,100]. Table 1 shows these values for each facility. The demand rates of the customers are also generated randomly using a uniform distribution over interval [0,20]. Moreover, the common service rate of the facilities is considered 30.

Table 1: Randomly generated parameters of the problem instances

Customer	μ_{j1}	μ_{j2}	${\sigma_j}^2$	λ_j
1	68.812	354.152	69.810	9.457
2	685.562	339.686	22.214	6.059
3	996.241	847.633	57.278	4.782
4	472.236	451.614	9.182	6.452
5	978.944	441.617	98.038	12.906
6	53.178	920.361	61.979	6.732
7	292.695	881.288	33.436	8.769
8	614.441	581.363	47.028	11.133
9	153.205	632.424	96.638	8.426
10	494.276	435.152	40.155	10.510

In Figure 6, the customers' mean locations and the candidate points within the convex hull of customers are shown, where 228 points are resulted by dividing the rectangle covering all customers using parallel lines with equal 50 units distance from each other. Among these points, 184 points are within the convex hull and considered to be the candidate points to locate the facilities.

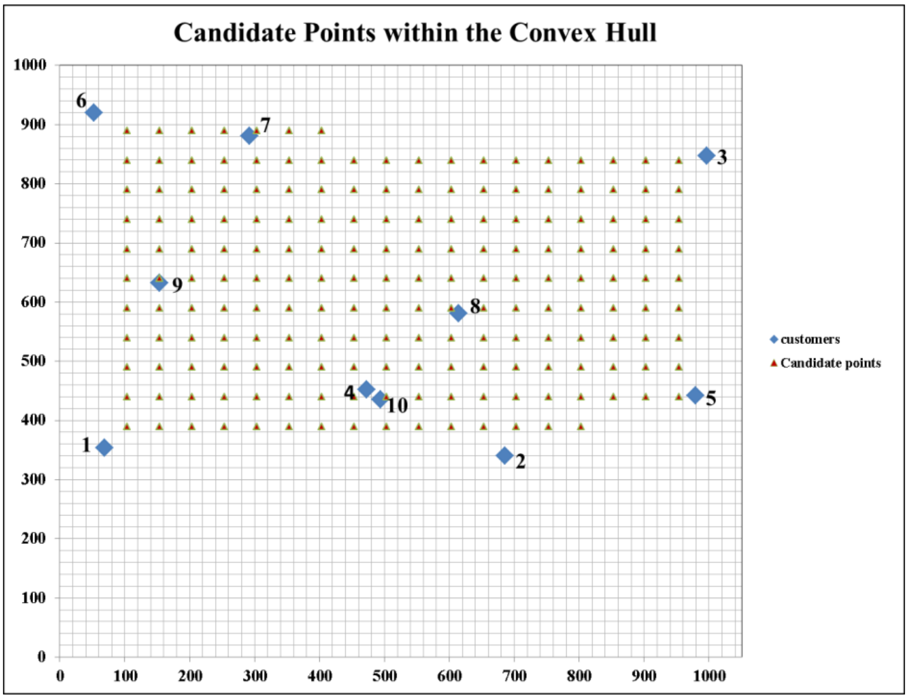

Figure 6: Customers' mean locations and candidate points for the discrete approximation of the problem

This problem is solved considering the first objective function through GA on a Core-i5 computer. The probability of choosing mutation operator is considered 0.5 and if chosen, each mutation operator is selected with an equal probability of 0.5. After 500 iterations, the results are shown in Table 2 as well as in Figure 7.

Table 2: The solution to the example problem considering the first objective function

Facility	Candidate point	x_i	y_i	Allocated customers
1	12	153.178	389.686	1
2	28	203.178	639.686	6,9,10
3	99	553.178	439.686	2,4
4	112	603.178	589.686	3,7,8
5	177	953.178	489.686	5
Objective Function		315.941	Time	4:54:1.141

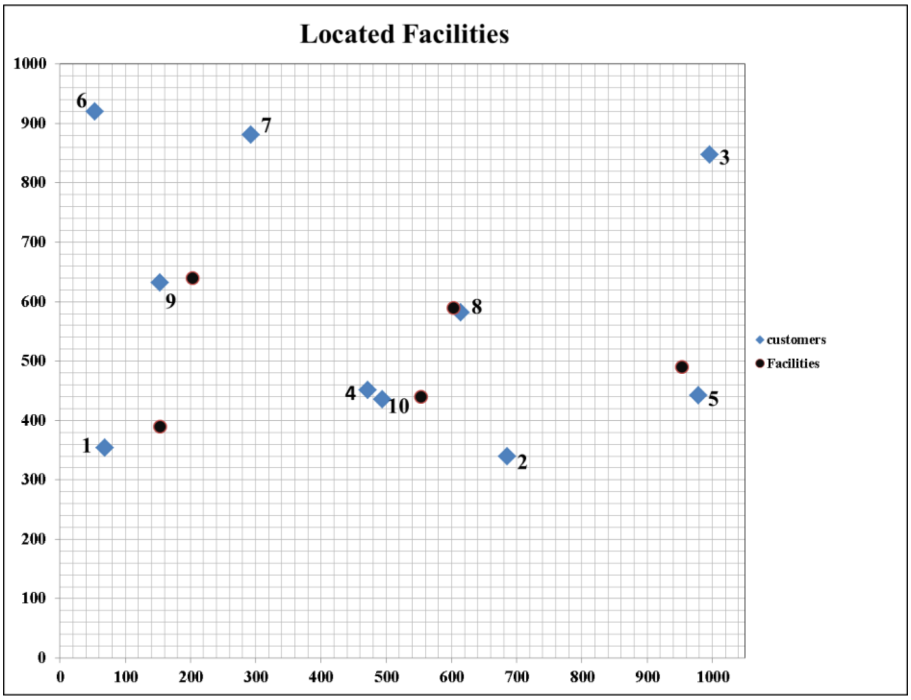

Figure 7: The resulted facility locations

The problem is also solved considering the second objective function separately. Besides, both NSGA-II and NRGA are utilized to find Pareto-optimal fronts for the bi-objective version of the problem. The results are shown in Table 3.

Table 3: The solutions of the single objective and multi-objective problems

Algorithm	First objective function	Second objective function	CPU time
GA	315.941	-	4:54:1.141
GA	-	0.909	4:26:22.304
NSGA-II	439.558	3.400	6:59:13.517
	457.478	2.234	
	471.451	1.969	
	475.105	1.677	
	494.685	1.567	
	502.224	1.367	
	599.936	1.235	
NRGA	434.360	2.311	7:38:34.689
	437.229	2.168	
	437.263	2.119	
	457.988	1.819	
	458.035	1.814	
	458.082	1.780	
	500.288	1.657	
	531.534	1.549	
	608.733	1.419	
	680.407	1.140	
	783.421	1.088	
	800.072	1.074	
	800.085	1.042	
	851.581	0.973	

The results in Table 3 indicate that the solutions found by NSGA-II are distributed properly in terms of extension and density of their distribution. Moreover, they are not so far from the single objective solutions. We note that decreasing one objective function results in an increase in the other one. In other words, none of the solutions dominate another and hence they belong to the Pareto-front found by the algorithm. These are also true for the solutions found by NRGA.

For a better comparison, we explore problems with various sizes. In Tables 4 & 5, the results for 7 instances are shown after 100 iterations. The parameters for these samples are produced using similar approaches as the ones described in the previous example. The computers used in this section are equipped with 1.7-GHz Processor and 1GB RAM, where run times are larger for larger problem sizes.

Table 4: The solutions to 7 different problems (1)

Problem	m	n	First objective function (GA)	Time	Second objective function (GA)	Time
1	5	3	221.524	0:18:20	0.533	0:19:13
2	7	4	291.778	0:26:37	0.881	0:32:24
3	12	8	733.289	2:10:52	1.300	2:54:52
4	12	8	616.030	2:6:34	3.054	2:31:45
5	15	8	650.673	3:12:20	4.125	3:51:44
6	15	12	866.011	3:33:27	3.208	3:41:54
7	20	15	1101.345	5:30:34	3.789	6:10:80

Table 5: The solutions to 7 different problems (2)

Problem	m	n	Number of solutions found by NSGA-II	Time	Number of solutions found by NRGA	Time
1	5	3	4	0:9:47	9	0:18:23
2	7	4	10	0:39:20	13	0:30:30
3	12	8	7	1:7:47	11	2:42:53
4	12	8	9	2:24:45	6	2:21:26
5	15	8	15	4:06:51	10	3:31:29
6	15	12	14	2:3:23	12	3:33:16
7	20	15	11	5:51:31	17	5:58:20

Various criteria such as the diversity and the convergence performance measures are defined to evaluate the performance of multi-objective evolutionary optimization algorithms [12]. In this problem, as the Pareto-Front is not known, we use the diversity metric denoted by Δ. It measures how evenly the points in the approximation set are distributed in the solution space. It can be calculated as follows:

$$\Delta = \sum_{i=1}^{|F_1|} \frac{|d_i - d|}{|F_1|}. \tag{20}$$

In Equation (20), d_i is the Euclidian distance between two consequent solutions in the first domination level and d is the mean of these values. This equation is only practical for problems with two objective functions. The deviation metric Δ can be calculated for each run of the algorithm, based on which an average of these deviations ($\bar{\Delta}$) is calculated over 10 runs in order to compare different algorithms. The algorithm with smaller $\bar{\Delta}$ value is more capable of spreading solutions in the obtained front.

In this paper, Δ is calculated in 10 runs of NSGA-II and NRGA, each run with 10 customers and 5 facilities (the parameters of these problems are produced exactly the same way as mentioned for the earlier problems). The results are shown in Tables 6 and 7. Note that the runtime in Table 6 is the least in Problem 9 because it was obtained using a faster computer.

Due to special characteristics of this problem, no exact solution exists. Firstly, the location of the demand is stochastically distributed for each customer; secondly, the demand occurrence time is also stochastic. Thus, there is no appropriate deterministic version of the problem that could be solved as a comparison solution. Consequently, the results of the above algorithms that are considered as the best feasible solutions found are compared with each other. The results in Tables 6 and 7 indicate that on one hand, NSGA-II generally performs better considering the number of solutions found, which provides the decision maker with more options. On the other hand, NRGA is better with respect to the deviation metric and gives a more diverse set of solutions.

Table 6: GA Solutions of 10 problems (m=10, n=5)

Problem	First objective function (GA)	Time	Second objective function (GA)	Time
1	522.491	-	2.949	1:49:10
2	458.985	1:27:15	5.467	1:54:56
3	617.971	1:31:29	2.752	2:13:13
4	574.023	1:41:16	1.471	1:45:46
5	638.965	1:30:28	4.053	1:40:55
6	668.993	1:36:19	2.356	1:49:24
7	423.302	1:43:56	2.472	1:55:49
8	552.024	1:28:13	2.672	1:53:90
9	455.075	0:59:29	1.701	1:10:33
10	500.097	1:13:40	1.689	1:26:50

Table 7: NSGA-II and NRGA solutions of 10 problems (m=10, n=5)

Problem	Number of solutions found by NSGA-II	Time	Δ	Number of solutions found by NRGA	Time	Δ
1	13	1:47:48	56.382	7	1:34:29	21.174
2	16	2:37:10	16.647	8	1:48:35	5.650
3	11	1:59:27	74.489	12	1:50:48	27.412
4	16	1:26:42	29.664	9	1:25:49	26.526
5	6	2:13:16	49.535	8	2:16:50	50.919
6	13	1:38:52	18.382	13	1:39:52	22.250
7	13	1:53:21	41.563	13	1:45:30	27.302
8	11	1:46:49	33.880	9	1:43:59	39.372
9	15	1:09:30	36.646	10	1:5:00	13.635
10	13	2:21:57	46.881	13	1:24:28	20.158
		$\bar{\Delta}$	40.407		$\bar{\Delta}$	25.439

5 Conclusion

In this paper, a new version of the multi-objective multi-facility location-allocation problem was discussed in which the locations of customers and their arrivals were probabilistically distributed. Uncertainty was taken into account in a stochastic form while the distance function was assumed to be Euclidian. Two mathematical models were proposed, a continuous one and a discrete approximation. Due to NP-hardness of the problem and the fact that simulation was needed to evaluate the objective functions of any feasible solution, finding optimal solutions of single-objective problems or Pareto-Front solutions of the multiobjective case was not practical. Consequently, a genetic algorithm was used to solve single-objective problems separately, while NSGA-II and NRGA were applied to solve the multi-objective problem. The results based on 18 test problems indicated that NSGA-II was the better approach in terms of the number of solutions found while NRGA performed better with respect to the diversity of the solutions obtained.

As this problem is not solvable optimally, making use of new tools such as big data-analysis techniques and advanced optimization methods such as distributed optimization are opportunities for possible improvements in solving such problems. Besides, another potential direction for future research could be to consider fixed costs for opening facilities. Here, one possible approach is to introduce a third objective to account for this component, as this type of cost is different from the other two objectives functions. Another approach is to consider a range of possible values for the number of facilities and optimize for each value (i.e. fixing the number of facilities and minimizing the two objective functions), assuming that building different facilities at various locations has approximately the same cost. Then the optimal number of facilities can be found. In addition, other distributions for the demand process and for the locations can be considered.

References

[1] Altınel, İ.K., Durmaz, E., Aras, N., and K.C. Özkısacık, A location-allocation heuristic for the capacitated multi-facility Weber problem with probabilistic customer locations, *European Journal of Operational Research*, vol.198, pp.790–799, 2009.

[2] Aras, N., Altınel, İ.K., and M. Orbay, New heuristic methods for the capacitated multi-facility Weber problem, *Naval Research Logistics*, vol.54, pp.21–32, 2007.

[3] Aras, N., Orbay, M., and İ.K. Altınel, Efficient heuristics for the rectilinear distance capacitated multi-facility Weber problem, *Journal of the Operational Research Society*, vol.59, pp.64–79, 2008.

[4] Brimberg, J., Hansen, P., and N. Mladenović, Decomposition strategies for large-scale continuous location-allocation problems, *IMA Journal of Management Mathematics*, vol.17, pp.307–316, 2006.

[5] Brimberg, J., Hansen, P., Mladenović, N., and S. Salhi, A survey of solution methods for the continuous location-allocation problem, *International Journal of Operations Research*, vol.5, pp.1–12, 2008.

[6] Brimberg, J., and N. Mladenović, Solving the continuous location-allocation problem with Tabu search, *Studies in Locational Analysis*, vol.8, pp.23–32, 1996.

[7] 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.

[8] Carrizosa, E., Conde, E., Munoz-Marquez, M., and J. Puerto, The generalized Weber problem with expected distances. *RAIRO-Recherche Operationnelle-Operations Research*, vol.29, pp.35–57, 1995.

[9] Cooper, L., Location-allocation problems, *Operations Research*, vol.11, pp.331–343, 1963.

[10] Cooper, L., Heuristic methods for location-allocation problems, *SIAM Review*, vol.6, pp.37–53, 1964.

[11] Deb, K., An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, vol.186, pp.311–338, 2000.

[12] Deb, K., Pratap, A., Agarwal, S., and T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol.6, pp.182–197, 2002.

[13] Drezner, Z., and S. Salhi, Incorporating neighborhood reduction for the solution of the planar p-median problem, *Annals of Operations Research*, 2017, doi: https://doi.org/10.1007/s10479-015-1961-y.

[14] Durmaz, E., Aras, N., and İ.K. Altınel, Discrete approximation heuristics for the capacitated continuous location-allocation problem with probabilistic customer locations, *Computers & Operations Research*, vol.36, pp.2139–2148, 2009.

[15] Farahani, R.Z., SteadieSeifi, M., and N. Asgari, Multiple criteria facility location problems: a survey, *Applied Mathematical Modelling*, vol.34, pp.1689–1709, 2010.

[16] Hajipour, V., Fattahi, P., Tavana, M., and D. Di Caprio, Multi-objective multi-layer congested facility location-allocation problem optimization with Pareto-based meta-heuristics, *Applied Mathematical Modelling*, vol.40, pp.4948–4969, 2016.

[17] Hosseininezhad, S.J., Salhi, S., and M.S. Jabalameli, A cross entropy-based heuristic for the capacitated multi-source Weber problem with facility fixed cost, *Computers and Industrial Engineering*, vol.83, pp.151–158, 2015.

[18] Jadaan, O.A., Rajamani, L., and C.R. Rao, Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NRGA, *Journal of Theoretical and Applied Information Technology*, vol.4, pp.61–68, 2008.

[19] Jamalian, A., and M. Salahi, Robust solutions to multi-facility Weber location problem under interval and ellipsoidal uncertainty, *Applied Mathematics and Computation*, vol.242, pp.179–186, 2014.

[20] Jiang, J.-L., Luan, F., Wang, L.-P., and X.-X. Zhu, A generalized Weber problem with different gauges for different regions, *Applied Mathematics and Computation*, vol.244, pp.244–260, 2014.

[21] Katz, I.N., and L. Cooper, An always convergent numerical scheme for a random locational equilibrium problem, *SIAM Journal of Numerical Analysis*, vol.11, pp.683–692, 1974.

[22] Katz, I.N., and L. Cooper, Normally and exponentially distributed locational equilibrium problems, *Journal of Research of the National Bureau of Standards*, Section: B, pp.53–73, 1976.

[23] Konak, A., Coit, D.W., and A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, *Reliability Engineering and System Safety*, vol.91, pp.992–1007, 2006.

[24] Logendran, R., and M.P. Terrell, Uncapacitated plant location-allocation problems with price sensitive stochastic demands, *Computers and Operations Research*, vol.15, pp.189–198, 1988.

[25] Love, R.F., and H. Juel, Properties and solution methods for large location-allocation problems, *Journal of the Operational Research Society*, vol.33, pp.443–452, 1982.

[26] Mladenovic, N., and J. Brimberg, A descent-ascent technique for solving the multisource Weber problem, *Yugoslav Journal of Operations Research*, vol.5, pp.211–219, 1995.

[27] Mousavi, S.M., Niaki, S.T.A., Mehdizadeh, E., and M.R. Tavarroth, The capacitated multi-facility location-allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms, *International Journal of Systems Science*, vol.44, pp.1897–1912, 2013.

[28] Mousavi, S.M., Sadeghi, J., Niaki, S.T.A., and M. Tavana, A bi-objective inventory optimization model under inflation and

discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO, *Applied Soft Computing*, vol.43, pp.57–72, 2016.

[29] Pasandideh, S.H.R., and S.T.A. Niaki, Genetic application in a facility location problem with random demand within queuing framework, *Journal of Intelligent Manufacturing*, vol.23, pp.651–659, 2012.

[30] Pasandideh, S.H.R., Niaki, S.T.A., and V. Hajipour, A multi-objective facility location model with batch arrivals: two parameter-tuned meta-heuristic algorithms, *Journal of Intelligent Manufacturing*, vol.24, pp.331–348, 2013.

[31] Rahmati, S.H.A., Hajipour, V., and S.T.A. Niaki, A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem, *Applied Soft Computing Journal*, vol.13, pp.1728–1740, 2013.

[32] Salhi, S., and M.D.H. Gamal, A GA-based heuristic for the multi-Weber problem, *Annals of Operations Research*, vol.123, pp.203–222, 2003.

[33] Sherali, H.D., and F.L. Nordai, NP-hard, capacitated, balanced p-median problems on a chain graph with a continuum of link demands, *Mathematics of Operations Research*, vol.13, pp.32–49, 1988.

[34] Taillard, É.D., Heuristic methods for large centroid clustering problems, *Journal of Heuristics*, vol.9, pp.51–73, 2003.

[35] Weber, A., *Uber den Standort der Industrien, Tubingen*, Translated as *Alfred Weber's Theory of the Location of Industries*, University of Chicago, Chicago, IL, 1909.

[36] Wesolowsky, G.O., The Weber problem with rectangular distances and randomly distributed destinations, *Journal of Regional Sciences*, vol.17, pp.53–60, 1977.

[37] Wesolowsky, G.O., The Weber problem: history and perspectives, *Location Science*, vol.1, pp.5–23, 1993.

[38] Zhou, J., Uncapacitated facility layout problem with stochastic demands, *Proceedings of the Sixth National Conference of Operations Research Society of China*, pp.904–911, 2000.

[39] Zhou, J., and B. Liu, New stochastic models for capacitated location-allocation problem, *Computers & Industrial Engineering*, vol.45, pp.111–125, 2003.