International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:6, 2011

A Bi-Objective Model for Location-Allocation Problem within Queuing Framework

Amirhossein Chambari, Seyed Habib Rahmaty, Vahid Hajipour, Aida Karimi

***Abstract*—**This paper proposes a bi-objective model for the facility location problem under a congestion system. The idea of the model is motivated by applications of locating servers in bank automated teller machines (ATMS), communication networks, and so on. This model can be specifically considered for situations in which fixed service facilities are congested by stochastic demand within queueing framework. We formulate this model with two perspectives simultaneously: (i) customers and (ii) service provider. The objectives of the model are to minimize (i) the total expected travelling and waiting time and (ii) the average facility idle-time. This model represents a mixed-integer nonlinear programming problem which belongs to the class of NP-hard problems. In addition, to solve the model, two metaheuristic algorithms including non-dominated sorting genetic algorithms (NSGA-II) and non-dominated ranking genetic algorithms (NRGA) are proposed. Besides, to evaluate the performance of the two algorithms some numerical examples are produced and analyzed with some metrics to determine which algorithm works better.

***Keywords*—**Queuing, Location, Bi-objective, NSGA-II, NRGA

I. Introduction

FACILITY location problems have been developed in the area of location-allocation (LA). LA models can be classified into discrete and continues models. The present study belongs to discrete LA models. Current et al. [10] present a short review of discrete location models. In this paper, we consider a bi-objective model for locating immobile (fixed) service systems with limited queue capacity. For locating facilities, the paper considers both customer and service provider perspectives simultaneously. Our perspectives are to limit (i) the time spent travelling and waiting at site and (ii) idle probability service facilities. For servicing these facilities, M/M/1/K queueing system is considered. It should be mentioned that the paper assumes customer travel only for open facilities. Li et al. [15] considered the optimal placement of web proxies in the internet and developed a dynamic programming algorithm to find the minimum cost location for them. A model for determining the optimal number and locations of proxy servers in a network is developed by Gautam [13]. His objectives were minimizing of setup/maintenance and operating costs.

Model formulation for optimal locating of switches in ATM communication networks based on quality of services was proposed by Marianov and Rl'os [16]. Hodgson [14] and Berman et al. [8] also independently proposed a closely related issue which is the flow-capturing model. The flow capturing problem can be applied in different terms such as locating gas stations, convenience stores [6]. According to Berman et al. [7] the stochastic queue median (SQM) model can be assumed for a mobile server such as an emergency response unit. Since in our problem there are many immobile server locations, the SQM problem is much different from our problem.

Berman et al. [4] developed a M/M/1/N queue model. Their objectives were: (i) the number of facilities treated by limiting the queue length, and (ii) the percentage of demand that may be lost because of model's limitations. They also studied numerical experimentation using nine heuristic approaches. Marianov and Serra [17], considering service congestion, proposed several probabilistic maximal location-allocation models with constrained waiting time for queue length. Studying the effect of using expected service time dependent queueing disciplines on the optimal location of a single server was done by Batta [3].

Berman et al. [5] attempted to locate optimally one server on a congested network and formulate a stochastic queue median by developing a heuristic algorithm. Meanwhile, Boffey et al. [9] reviewed congestion model with immobile server in terms of location of facility. Pasandideh and Niaki [18] presented a bi-objective model for facility location problem in M/M/1 and solved it with traditional multiobjective genetic algorithm. However, this paper not only considers M/M/1/K model (instead of M/M/1 model) but also uses new improved evolutionary algorithms (NSGA-II and NRGA) to solve problems much more effectively. A Fuzzy location-allocation model for congested systems was studied by Shavandi and Mahlooji [20]. Utilizing fuzzy theory, they developed a queuing maximal covering location-allocation model and called it "the fuzzy queuing maximal covering location-allocation model".

Wang et al. [21] proposed a model that is used as the core of our paper. Their model considered facility location problems within framework of M/M/1and patronized customers to the closest open facility. Nevertheless, their perspective in their model has only considered customer (but not the service provider). Three approaches including developing (i) an efficient Lagrangian relaxation procedure,

Amirhossain Chambari is with Islamic Azad University, Qazvin Branch, Qazvin, Iran. (phone: +989163000753; fax: +982813358026; e-mail: amir.chambari@gmail.com).

Seyed Habib Rahmatyis with Islamic Azad University, Qazvin Branch, Qazvin, Iran. (phone: +989163000753; fax: +982813358026; e-mail: sd_rahmaty@yahoo.com).

Vahid Hajipour is with Islamic Azad University, Qazvin Branch, Qazvin, Iran. (phone: +989123829766; fax: +982813358026; e-mail: v.hajipour@yahoo.com).

Aida Karimi is with Islamic Azad University, Qazvin Branch, Qazvin, Iran. (phone:+989123829766;fax:+982813358026; e-mail: aida.karimi59@yahoo.com).

(ii) a tabu search algorithm and (iii) a greedy-dropping heuristic was investigated in their paper.

The previous papers have pursued only one objective function that differs among: (i) customer perspective, (ii) service provider perspective or (iii) combination of the two mentioned perspectives. In this paper, firstly, we propose a bi-objective model for facility location problem with stochastic customer and immobile servers. Secondly, in order to solve the model, we propose two improved metaheuristics: (i) NSGA-II and (ii) NRGA.

This paper is organized as follows. In section 2, the problem and the mathematical formulation of our model is defined precisely. In section 3, the applications of two metaheuristic algorithms including NSGA-II and NRGA are described. Section 4 is devoted to the computational experiment and the analysis of the results. Finally, some conclusions are presented in Section 5.

II. Problem definition

The ATM and internet mirror sites are examples of the system under consideration. For the system under study here in this paper, the decision maker encountered with two objectives, both of which are going to be minimized. The objectives are: (i) total of the expected travelling and waiting customers, and (ii) the average facility idle-time probability. The model should determine the optimal allocations of some business canters (customers) to the machines (service provider) with respect to objectives and constraints.

In order to model the problem, the paper considers the following general assumptions: travel to the facilities for obtaining the service. Each of the facilities hosts a single server. Each of the customer nodes only can be allocated to one server. Requests for service from each of the customer nodes follow an independent Poisson stream. Each of the open facility sites has only one server with exponential service times. Fixed cost for each open potential facility sites is assumed. A fixed cost is assumed per each unit of increasing to the capacity of the opened queue facility j . A total cost for locating and determining capacity of system is assumed. Maximum number of facilities that can be opened is assumed. For system capacity of each potential facility an upper and lower bound is assumed. There is no information about the status of the facilities for customers. The parameters and the variable of the model associate with proposed mathematical model are as follows:

H upper bound of system capacity of potential facility j

h lower bound of system capacity of potential facility j

M number of customer nodes

N number of potential service nodes

Q maximum number of facilities that can be opened ($Q \le N$)

t_{ij} the travelling time matrix from customer node i to facility node j

λ_i demand rate of service requests from customer node i

μ_j common service rate of each server j

γ_j demand arrival rate at open facility site j

w_j expected waiting time of customers assigned to facility node j

f_{1j} fixed cost of locating a facility at site j

f_{2j} cost of one unit increase to the capacity of system of opened facility j

C cost constraint

r_j demand customer rate divided to service rate at open facility j

π_{0j} probability of the idle-time of the server at opened facility (idle probability) j

z_1 total of travelling and waiting time of all costumer nodes

z_2 average idle probabilities of all facilities (service provider)

y_j 1 if a facility is opened at node j and otherwise 0

x_{ij} 1 if customer i is assigned to facility j and otherwise 0

k_j system capacity of the facility j which is opened.

Then, the aggregate travelling time of the customers per unit time can be obtained by:

$$v_1 = \sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i t_{ij} x_{ij} \quad (1)$$

Since an opened facility behaves as M/M/1/K model where K the capacity of the system is (Gross and Harris [12]), its expected waiting time is calculated as:

$$w_j = \frac{\left(\frac{r_j}{1-r_j} - \frac{(k_j+1)r_j^{k_j+1}}{1-r_j^{k_j+1}}\right)}{\gamma_j\left(1-r_j^{k_j}\frac{1-r_j}{1-r_j^{k_j+1}}\right)} \quad (2)$$

Where $\gamma_j = \sum_{i=1}^{M} \lambda_i x_{ij}$, $r_j = \frac{\gamma_j}{\mu_j}$.

Notice, since in M/M/1/K model utilization rate is formulated as $\rho_j = \frac{\gamma_j(1-\pi_{kj})}{\mu_j}$ (where π_{kj} shows the probability that K costumers are inside the system), r_j and ρ_j are not the same notations. The aggregate waiting time of customers per unit time is calculated as:

$$v_2 = \sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i w_j x_{ij} = \sum_{j=1}^{N} \frac{\left(\frac{r_j}{1-r_j} - \frac{(k_j+1)r_j^{k_j+1}}{1-r_j^{k_j+1}}\right)}{(1-r_j^{k_j})\frac{1-r_j}{1-r_j^{k_j+1}}} \quad (3)$$

Thus, the first objective is the sum of the travelling and waiting time ($z_1 = v_1 + v_2$) which is going to be minimized. According to the characteristics of a M/M/1/K model, the idle probability at open facility j is calculated as:

$$\pi_{0j} = \frac{1-r_j}{1-r_j^{k_{j+1}}} \quad (4)$$

On the other hand, the second objective is to minimize z_2 as the average of π_{0j} for all opened facilities j.

The mathematical formulation of the model is as follows:
Basic model:

$$Min\ Z_1 = \sum_{i=1}^{M}\sum_{j=1}^{N} \gamma_j t_{ij} + \sum_{j=1}^{N} \frac{\left(\frac{r_j}{1-r_j} - \frac{(k_j+1)r_j^{k_j+1}}{1-r_j^{k_j+1}}\right)}{(1-r_j^{k_j})\frac{1-r_j}{1-r_j^{k_j+1}}} \quad (5)$$

$$Min\ Z_2 = \frac{\sum_{j=1}^{N} \pi_{0j}}{\sum_{j=1}^{N} y_j} \quad (6)$$

s.t

$$hy_j \le k_j \le Hy_j \qquad \forall j \in N \quad (7)$$

$$\sum_{j=1}^{N} y_j \le Q \quad (8)$$

$$\sum_{j=1}^{N} x_{ij} = 1 \quad \forall i \in M, \quad (9)$$

$$x_{ij} \le y_j \quad \forall i \in M, j \in N, \quad (10)$$

$$\sum_{i=1}^{M} \lambda_i x_{ij} \le k_j \quad \forall j \in N, \quad (11)$$

$$\gamma_j = \sum_{i=1}^{M} \lambda_i x_{ij} \qquad \forall j \in N, \quad (12)$$

$$r_j = \frac{\gamma_j}{\mu_j} \quad \forall j \in N, \quad (13)$$

$$\pi_{oj} = \frac{1-r_j}{1-r_j^{k_j+1}} y_j \quad \forall j \in N, \quad (14)$$

$$\sum_{j=1}^{N} f_{1j} y_j + \sum_{j=1}^{N} f_{2j} k_j \le C \quad \forall j \in N, \quad (15)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in M, j \in N, \ y_j \in \{0,1\} \ \forall j \in N, \quad (16)$$

$$k_j \ge 0, \text{ integer } \forall j \in N \quad (17)$$

The model aims to minimize two objectives: (i) the total of expected travelling and waiting time customers and (ii) average facility idle-time. These two objectives are conflicting; it means that an improvement in the value of one of them leads to no change or aggravating of the value of the other one.Constraint (7) assures that system capacity at the open facility j falls between upper bound and lower bound, Constraint (8) sets an upper bound for the maximum number of opened facilities. Constraints (9 and 10) are used to show each customer demand is satisfied by only one opened facility. Constraint (11) assures that the input to each server to be less than its capacity of queue. Constraint (12) calculates the input of each server.Constraint (13) calculates the server's ratio demand divided to service's ratio for each servers. Constraint (14) calculates the probability of idle time for each severs. Constraint (15) represents the available cost. Finally, Constraints (16) and (17) represent the ranges of the model variables.

Above depicted model is the basic model and can be simplified in the statues that $r_j = 1$. In this situation w_j (Eq. 2) is simplified to $w_j = \frac{k_j}{2\gamma_j(\frac{k_j}{k_j+1})}$. Consequently, the aggregate waiting time of customers per unit time (V_2, Eq.3) is simplified to $T_2 = \sum_{j=1}^{N} \frac{(k_j+1)}{2}$ and π_{oj} (Eq.4) is simplified to $\pi_{oj} = \frac{1}{k_j+1}$.

What worsens our problem is: (i) the mixed integer nonlinear programming model (which is assign to the NP-hard class of problems), and (ii) the large number of constraints. The exact techniques for solving the problem are not necessarily desirable since it is very hard to obtain exact solutions. Consequently, in this paper, two improved metaheuristic algorithms for accessing near optimality Pareto solutions are used.

III. Multi Objective Metaheuristic Algorithms

In this paper, two multi-objective genetic algorithms, including non-dominated sorting genetic algorithms (NSGA-II) and non-dominated ranking genetic algorithms (NRGA) are proposed. The two algorithms are illustrated more precisely as follows:

A. Non-dominated sorting genetic algorithm (NSGAII)

NSGA-II is one of the most efficient and well-known multi-objective evolutionary algorithms proposed by Deb et al. [11]. Its structure is described in Fig.1. For clarity reasons, the different steps of the algorithm are explained in details.

1. *Initialization*

To represent the parameters of the algorithms, the following notations are used:
Population size (*popsize*): It is the number of the individuals or scenarios that will be kept in each generation. Crossover rate (P_c): It is the probability of performing a crossover in the

multi-objective genetic algorithm. Mutation rate (P_m): It is the probability of performing mutation in the multi-objective genetic algorithm.

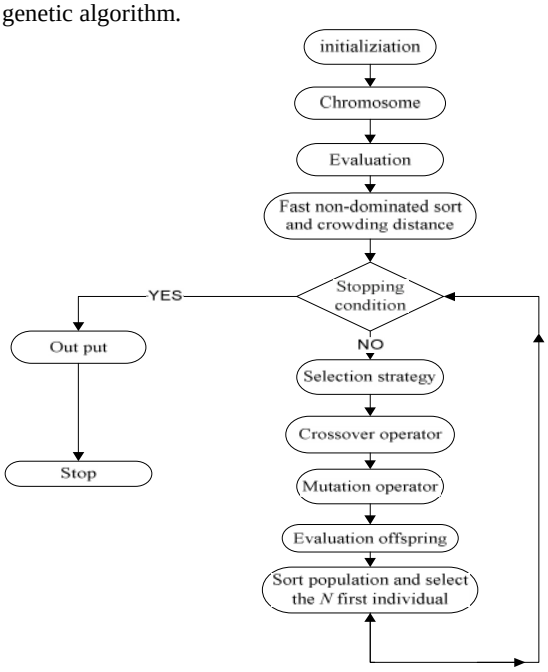

Fig.1. Flowchart of NSGA-II

2. *Chromosome Evaluation*

Representation is one of the most important issues in improving performance of genetic algorithms. The encoding process of solution is made up of four parts. The first part is a continuous string with length of customer nodes whose genes are filled by random numbers between 0 and 1. The second part is similar to the first part but its length is the same as the amount of potential facility nodes. The third part is similar to the second part but, its genes vector are filled with uniform random distribution numbers between lower limit system capacity and upper limit system capacity. Finally, the fourth part of chromosome is a random number between 1 and Q. Chromosome structure is shown in Fig.2 below.

1-	a_1	a_2	a_3	…	a_i	a_{m-1}	a_M
2-	b_1	b_2	b_3	…	b_j	b_{n-1}	b_N
3-	k_1	k_2	k_3	…	k_j	k_{N-1}	k_N
4-	$1 \le q \le Q$						

Fig.2 Structure of a chromosome

3. *Decoding of the Chromosome*

As we mentioned in the previous section, the chromosome which is decoded in a backward direction is illustrated through an example. Suppose eight customer nodes and six potential facility nodes exist and the maximum number of opened facilities (Q) and upper (H) and lower (h) bound of system capacity for opened facilities are four, thirty eight and twenty, respectively.

Firstly, according to the forth part of the chromosome, a random number between one and four is generated (assume it is three). Then according to the second part of chromosome, a vector with six genes is produced and sorted ascendingly. Next, the first three genes of the sorted vector are selected to represent opened facilities location. So, each gene illustrates the number of potential position of facilities before sorting process.

Then, similar to the first part of chromosome, a vector of the same size as the customer nodes with random number is generated. In order to allocate customer nodes to opened facilities, first, integer part of each gene ($p_i = \lfloor q \times b_i + 1 \rfloor$) that is a random integer number between 1 to q) is computed. Then if $0 \le b_i \le \frac{1}{3}$, p_i becomes one (Customer i is allocated to facility 5) else If $\frac{1}{3} \le b_i \le \frac{2}{3}$ p_i becomes two (customer i is allocated to facility 2) and finally if $\frac{2}{3} \le b_i \le 1$ p_i becomes three (customer i is allocated to facility 6).

Finally, like the third part of the chromosome, after generating a vector of the same size as the amount of potential facilities, by applying $\lfloor (H - h + 1) \times k_j \rfloor + h$ for each of its genes a number between h and H is obtained. Now, if a potential facility is opened, its system capacity will be considered (i.e. its system capacity will be assigned). Otherwise, it will become zero. Chromosome decoding is shown in Fig.3.

1-	0.98	0.1	0.54		0.56	0.04	0.23
2-	0.04(5)	0.1(2)	0.23(6)		0.54(3)	0.56(4)	0.98(1)
3-	0.43(2)	0.1(1)	0.78(3)	0.17(1)	0.54(2)	0.32(2)	0.13(1)	0.23(1)
4-	38	24	22	29	32	26	

Fig.3 Decoding representation.

4. *Chromosome Evaluation*

In a multi-objective GA approach, as soon as a chromosome is generated, it needs a fitness value. But for evaluating the obtained chromosomes and assigning a fitness value to them, the chromosomes should be decoded. The process of decoding was illustrated in the previous sub section.With respect to the model's constraints, we don't expect a random produced chromosome being feasible. Thus, the main problem our multi-objective genetic algorithms face is how to deal with constraints. To do so, penalty methods as one of the first approaches in GA (Yeniay and Ankare [22]) are used. Penalty functions were used in their paper to penalize infeasible solutions by reducing their fitness values in proportion to the degree of their violation. The penalty method transforms a constrained problem to an unconstrained one as illustrated below. In this paper, the additive penalty function is used as shown in Eq .18.

$$F(\bar{x}) = \begin{cases} f(\bar{x}), & \bar{x} \in \text{feasible region} \\ f(\bar{x}) + p(\bar{x}), & \text{otherwise} \end{cases} \quad (18)$$

That $p(\bar{x})$ is penalty's value. If no violation occurs, $p(\bar{x})$ will be zero; otherwise, it will take a positive amount. Since different constraints may differ in different dimensions, it is essential to normalize all constraints. After normalizing, constraints, violations find the same dimension and the total violation can be computed as the sum of all normalized constraint violations. For instance, a constraint like $g_j(x) \le b_j$ can be normalized as $\frac{g_j(x)}{b_j} - 1$. With regard to the presented chromosomes of model, only constraints 11 and 15 may be violated. In such a situation these constraint are normalized and formulated as Eq.18.

5. *Fast Non-Dominated Sort and Crowding Distance*

Ranking of the population is done within following subsection:

6. *Fast non-dominated sort*

Ranking of population is done by using the concept of domination. The pseudocode for finding nondominated members in a population (P_o) is shown in Table 1.

Table 1
Fast-non dominated sort (Po) pseudocode algorithm

for each $p \in Po$	
$S_p = \emptyset$	
$n_p = 0$	
For each $q \in Po$	
If $(po \prec q)$ then	if po *dominates* q
$S_p = S_p \cup \{q\}$	add q to the set of solutions dominated by add po
Else if $(q \prec po)$ then	
$n_{po} = n_p + 1$	increment the domination counter of po
If $n_p = 0$ then	po belong to the first front
$po_{rank} = 1$	
$f_1 = f_1 \cup \{po\}$	
$i = 1$	initialize the front counter
While $f_i = \emptyset$	
$Q = \emptyset$	used to store the members of the next front
For each $po \in f_i$	
For each $q \in S_{Po}$	
$n_q = n_q - 1$	
If $n_q = 0$ then	q belongs to the next front
$q_{rank} = i + 1$	
$Q = Q \cup \{q\}$	
$i = i + 1$	
$f_i = Q$	

7. *Crowding Distance*

The Density of solutions is illustrated by a measure called crowding distance. Crowding distance gives an estimate of the density of solutions which are laid surrounding a particular solution in the population. The average distance of two points on either side of this point along each of the objectives is evaluated by such measures Deb et al. [11].The pseudocode of calculating the crowding distance is shown in table 2.

Table 2
Crowding-distance-assignment(τ) algorithm pseudocode

$G = \lvert n \rvert$	number of solutions in n
For each j, set $\tau[j]_{distance} = 0$	initialize distance
For each objective m	
$n = sort(\tau, m)$	sort using each objective value
$n[1]_{distance} = n[G]_{distance} = \infty$	so that boundary points are always selected
For $j = 2$ *to* $(G-1)$	for all other points
$n[j]_{distance} = n[j]_{distance} + ([j+1].m + [j-1].m)/(f_m^{max} - f_m^{min})$	

8. *Stopping condition*

There is no standard ending condition for stopping multi-objective genetic algorithms. In this paper, the metaheuristics algorithms are simply stopped after a fixed number of iterations.

9. *Selection strategy*

In this section, tournament selection operator is used for selecting k individuals for the next generation. To do so, crowding distance and nondominated rank are used as the criteria for guiding the selection process. In this process, first ranking operator should be calculated. Then, for the solutions with same rank crowding distance operator is calculated. The process can also been checked Table3.

Table 3
crowded comparison operator algorithm pseudocode

$i \prec j$ if $(i_{rank} < j_{rank})$ or $(i_{rank} = j_{rank})$ and $(i_{distance} > j_{distance})$

10. *Crossover operator*

The crossover operator is a very important operator in MOGA. The main idea behind crossover is that a combination between segments of two individuals might yield a new individual which benefits from both parents' advantages. The crossover operator gets two individuals as its input, and gives two new individuals (i.e. the offspring).

In this paper, using a user-determined crossover probability (P_c), the crossover operator is a kind of linear combination of the two parents that uses the following equations for each gene:

$$offspring_1 = parent_1 - \beta \times (parent_1 - parent_2) \quad (19)$$

$$offspring_2 = parent_2 - \beta \times (parent_1 - parent_2) \quad (20)$$

Let β be a random value between 0 and 1.

11. *Mutation operator*

Mutation is another major genetic operator that is used to preserve genetic diversity from one generation of a population to the next iterations. The mutation operator explores those solution spaces which are not explored by crossover operator. In this paper, random mutation operator is utilized. To do so, a random selected gene exchanges with a random value within the range of the gene's minimum and maximum value.

12. *Evaluation offspring*

After using of crossover and mutation operators, the fitness function can be calculated.

13. *Sort population and select the N first individual*

First, Parent (P_t) and offspring (Q_t) population should be combined ($R_t = P_t \cup Q_t$). It means that the population R_t comes from two populations. All population members are included in R_t . This can strongly ensure elitism of the algorithm. Then, all solutions of R_t are sorted based on their nondomination status. If the total number of solutions belonging to the best non-dominated set F_1 is smaller than N , F_1 is added into $P_{(t+1)}$. In the order of their ranking, the remaining members of the population $P_{(t+1)}$ are selected from the next nondominated fronts. To select exactly N . solutions, the solutions of the last included front are sorted and then filled in the available slots in $P_{(t+1)}$, using the crowded comparison operator; the best solutions (i.e., those with larger values of the crowding distance) are selected.

B. Non-dominated sorting genetic algorithm (NSGAII)

NRGA was introduced by Al jadaan et al. [1]. But, In contrast to the NSGA-II, the difference between the
NRGA and the NSGA-II is their different selection strategy (segment 4.1.6). In NRGA, instead of binary tournament selection, roulette wheel selection is utilized. Al jadaan et al. [2] applied roulette wheel selection algorithm. In that algorithm, a fitness value equal to its rank in the population is assigned to each individual.

First, sort population according to fast non-domination sorting and choose the best solutions from the first ranked population. Then, according to their crowding distance criteria, individuals of each front are ranked.

Now, two tiers ranked based roulette wheel selection are used (one tier to select the front and the other to select solution from the front).

The front probability obtained as Eq. (21).

$$P_i = \frac{2 \times rank_i}{N_F \times (N_F + 1)} \quad \forall, i = 1...N_F \tag{21}$$

Where N_F show the number of fronts. In this equation, it is obvious that a front with highest rank has the highest probability to be selected. So the probability of individuals fronts based on their crowding distance criteria is calculated as follows:

$$P_{ij} = \frac{2 \times rank_{ij}}{M_i \times (M_i + 1)} \quad \forall, i = 1...N_F \ \forall, j = 1...M_i \tag{22}$$

Where M_i show the number of individuals in the front i . In this equation individuals with more crowding distance have more selection probability. The diversity among non-dominated solutions is also considered. Next, roulette wheel selection is applied according to the two random numbers (indicate number of front and individual chromosome in selected front) in intervals [0, 1] and [0, 1] respectively. This process is repeated until the desired number of individuals has been selected.

IV. Computational Experience

In this section, several test problems, with different sizes, are solved to evaluate the performance of the two presented metaheuristic algorithms. The metaheuristics algorithms are coded and compiled with matlab (R2009a) and the computational experiment is performed on a PC with a Pentium 1860 processor and 1 GB RAM.

The experiments are implemented on three different sizes of test problems which differs among small size (summation of numbers of customer zones and potential facility sites equals 15), medium size (summation of numbers of customer zones and potential facility sites equals 30), and large size (summation of numbers of customer zones and potential facility sites equals 45) were generated randomly to have a wide range of problem structures. For each size, five-test problems were generated randomly. Customer zones and potential facility sites were generated from a uniform distribution over a square with side 100. The demand requirements of the customers were drawn from a uniform distribution between 40 and 80. The service rate of each server was drawn from a uniform distribution between 50 and 10. The travelling time t_{ij} is computed as being a proportion of the Euclidean distance among the location of customers and potential facilities. Fixed setup costs of locating and cost of adding one unit to system's capacity are related to service rate of each size of test problems.

A. Parameter tuning

Since the results of all metaheuristics techniques are sensitive to their parameter setting, it is required to do extensive simulations to find suitable values for various parameters. The parameters of the two proposed metaheuristics algorithm are *popsize*, P_c , P_m *and MaxGen*.

The parameters of the two proposed metaheuristics algorithm are regulated using a design of experiment (DOE) approach. To achieve this aim using DOE, we carried out extensive experiments to determine effective parameters.

In order to execute the procedure, we used MINITAB software for finding the relation between responses (objective functions) and effective factors on responses (*popsize*, P_c , P_m *and MaxGen*). Finally, the optimum values of the parameters of both algorithms are obtained and presented in Tables 4 and 5. In addition, the adjusted values of parameter cause more effective Pareto solutions.

Table 4
NSGA-II parameter results

parameters	Small	Medium	Large
popsize	93	130	200
P_c	0.9	0.8	0.8
P_m	0.05	0.12	0.3
MaxGen	100	170	250

Table 5
NRGA parameter results

Parameters	Small	Medium	Large
Popsize	100	134	189
P_c	0.9	0.82	0.9
P_m	0.05	0.3	0.6
MaxGen	100	178	270

B. Performance Measures

Due to the conflicting nature of Pareto curves, we should use some performance measures to have a better assessment of multiobjective algorithms. So the following four performance metrics are considered:

The first metric is the number of Pareto solution (NPS), which shows the number of Pareto optimal solutions that each algorithm can find. The second metric is Set Coverage Metric introduced by Zitzler and Thiele [23].This measure compares two Pareto-optimal sets with each other. Given two Pareto optimal sets A and, the coverage $C(A,B)$ is calculated as:

$$C(A,B)=\frac{|\{b\in B\mid \exists a\in A: a\succ b\}|}{|B|} \quad (23)$$

Where $|B|$ means the number of solutions in the set B, and $a\succ b$ means that solution a weakly dominates solution b, i.e., the objectives of a are both less than those of b. Therefore, $C(A,B)$ tells the fraction of B weakly dominated by A.

Note that $C(A,B)\neq 1-C(B,A)$.

Note: In order to evaluate how much the coverage metric of one algorithm is better than the other one, their coverage is normalized by the Eq. 24.

$$Q(A,B)=\frac{C(A,B)}{C(A,B)+C(B,A)} \quad (24)$$

Note that $Q(B,A)=1-Q(A,B)$.

The Third metric is Spacing metric (*S*) introduced by Schott in [19].The metric *S* measures the extent of spread achieved among the obtained solutions.

This metric is given by:

$$S=\sqrt{\frac{1}{|n|}\sum_{i=1}^{n}(d_i-\bar{d})^2} \quad (25)$$

Where $d_i=\underbrace{\min}_{k\varepsilon n\wedge k\neq 1}\sum_{m=1}^{2}|f_m^i-f_m^k|$, $\bar{d}=\sum_{i=1}^{n}\frac{d_i}{|n|}$, is the mean of all d_i and n is the size of the known Pareto front.

The fourth metric is computational time of the algorithm (CPU) which indicates the computational time of each metaheuristic algorithm. The results of experiments and comparisons of metaheuristic algorithms for their different sizes are presented in Table 6.

C. Results

In this section, we discuss our results and investigate the effectiveness of the two proposed metaheuristics algorithms. To evaluate the performance of the mentioned algorithms, they are executed on different sizes of test problems. For each group of problems (which are classified according to their size), four metrics are calculated and the results are shown in Table 6. It should be mentioned that each group size consists of five test problems. Then the average amounts of outputs are calculated as the determining criteria of the paper. Outputs show that for the different size test problems, NSGA-II is the best criterion, but the computational time of NRGA is better than NSGA-II. Fig. 3 presents the non-dominated solutions of a single run of the proposed algorithms (NSGA-II and NRGA).

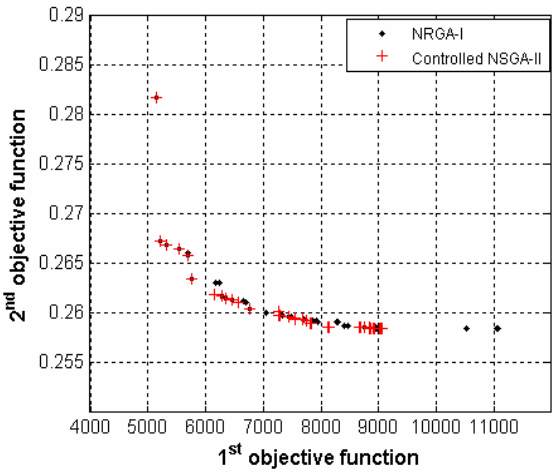

Fig.4 An example for non-dominated solutions problem

V. Conclusion and Future Research

Most real-world engineering problems consist of different objectives simultaneously. Therefore, a trade-off among these objectives is very important for us. In this paper, we examined the bi-objective problem of facility location problem with stochastic customer demand and immobile servers within M/M/1/K queue model. The model aims to determine the best site of facilities; the best strategy for assigning customers to opened facilities, and determine system capacity of opened facilities. The paper has developed a nonlinear mixed integer-programming model and two metaheuristics algorithms (NSGA-II and NRGA) executed them for the produced test problems. Four quantitative performance metrics were used to analyze the diversity and convergence of algorithms. Finally, the outputs revealed that NSGA-II satisfy the criterion better than NRGA.

The following approaches can be proposed to the future researchers:

- Considering another objective function (e.g. cost objective function) instead of average idle objective function
- Developing a chromosome representation that does not require penalty function
- Considering random or fuzzy parameter for the problem.
- Considering other MOEAs such as MOPSO or MOSA for solving the problem.
- Developing other queuing system rather than M/M/1/K.
- Developing of heuristic approach instead of generating random data in the initial segment.

REFERENCES

[1] O. Al Jadaan, C. R. Rao, L. Rajamani, non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NRGA. Journal of Theoretical and Applied Information Technology, 2, 60-67, 2008.
[2] O. Al Jadaan, C. R. Rao, L. Rajamani, improved selection operator GA. Journal of Theoretical and Applied Information Technology, 2, 269-277, 2008.
[3] R. Batta, J.M. Dolan, N. N. Krishnamurthy, The maximal expected covering location problem: Revisited. Transportation Science, 23, 277–287, 1989.
[4] O. Berman, D. Krass, J.Wang, Locating service facilities to reduce lost demand. IIE Transactions, 38, 933 – 946, 2006.
[5] O. Berman, The maximizing market-size discretionary facility location problem with congestion. Socio-Economic Planning Sciences, 29, 39–46, 1995.
[6] O. Berman, M. J. Hodgson, D. Krass, Flow-interception problems, in: Facility Location: A Survey of Applications and Methods, ed. Z. Drezner, Springer Series in Operations Research, 1995.
[7] O. Berman, R.C. Larson, S.S. Chiu, Optimal server location on a network operating as an M/G/1Queue. Operations Research, 33, 746–771, 1985.O. Berman, R.C. Larson, N. Fouska, Optimal location of discretionary service facilities. Transportation Science, 26, 201–211, 1992.
[8] B. Boffey, R. Galvao, L. Espejo, A review of congestion models in the location of facilities with immobile servers.European Journal of Operational Research, 178, 643–662,2007.
[9] J. Current, M. Daskin, D. Schilling, discrete network location models, in: drezner, z., hamacher, h.w., (eds.), facility Location: applications and theory. Springer, Heidelberg, 14, 80-118, 2002.
[10] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-Objective Optimization: NSGA-II. In: Proceedings of the parallel problem solving from nature VI (PPSN-VI) conference, 849-858, 2000.
[11] D. Gross, C. M. Harris, Fundamental of queuing theory (3nd ed.). New York, NY: Wiley-Interscience, 1998.
[12] N. Gautam, Performance analysis and optimization of web proxy servers and mirror sites. European J Oper Res, 142, 396–418, 2002.
[13] M.J. Hodgson, A flow-capturing location-allocation model, geographical analysis, 22, 270 - 279, 1990.
[14] B. Li, , M.J. Golin, G. F. Italiano, X. Deng, K. Sohraby, On the optimal placement of web proxies in the Internet. Proc INFOCOM, 99, 1282–1290, 1999.
[15] V. Marianov, M. Rios, A probabilistic quality of service constraint for a location model of switches in ATM Communications networks. Ann Oper Res 96, 237–243,2000.
[16] V. Marianov, d. Serra, Probabilistic maximal covering location-allocation for congested system. Journal of Regional Science, 38, 401–424, 1998.
[17] S. H. R. Pasandideh, S. D. A. Niaki, Genetic application in a facility location problem with random demand within Queueing framework. J Intell Manuf, 21, 269-278, 2010.
[18] J. R. Schott, Fault tolerant design using single and multicriteria genetic algorithms optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1995.
[19] H. Shavandi, H. Mahlooji, A fuzzy queuing location model with a genetic algorithm for congested systems. Applied Mathematics and Computation, 181, 440–456, 2006.
[20] Q. Wang, R. Batta, C. M. Rump, Algorithms for a facility location problem with stochastic customer demand and immobile Servers. Annals of operations Research, 111, 17-34, 2002.
[21] O. Yeniay, B. Ankare, Penalty function methods for constrained optimization with genetic algorithms.Mathematical and computational application,10, 45-56, 2005.
[22] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms a comparative case study. In A. E.Eiben, T. Back, M. Schoenauer and H. P. Schwefel (Eds.), Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Berlin, Germany, 292 – 30, 1998.

TABLE VI
RESULTS OF THE EXPERIMENT OF DIFFERENT SIZE TEST PROBLEMS

M	N	**NSGA-II**				**NRGA**			
		Q [%]	S	NPS	CPU [seconds]	Q [%]	S	NPS	CPU [seconds]
10	5	0.41	36.14	51	297.07	0.59	25.09	53	306.65
7	8	0	86.36	45	244.94	1	154.97	45	246.05
9	6	0.93	4.17	62	223.23	0.07	10.41	60	243.91
8	7	0.49	78.93	73	269.65	0.51	24.93	67	280.70
6	9	0.3	3.1	80	265.23	0.7	2.97	80	273.18
20	10	1	36.66	70	408.38	0	87.22	50	405.38
15	15	1	8.67	80	425.37	0	23.85	70	415.63
17	13	0.86	13.87	73	443.37	0.14	45.17	74	454.45
10	20	1	20.77	73	479.49	0	30.1	83	447.66
22	8	1	44.48	77	442.64	0	183.05	29	436.47
20	10	1	36.66	70	408.38	0	87.22	50	405.38
15	15	1	8.67	80	425.37	0	23.85	70	415.63
17	13	0.86	13.87	73	443.37	0.14	45.17	74	454.45
10	20	1	20.77	73	479.49	0	30.1	83	447.66
22	8	1	44.48	77	442.64	0	183.05	29	436.47
Average		0.79	30.50	70.46	379.90	0.29	63.81	61.13	377.97