A Cross Entropy-Based Heuristic for the Capacitated Multi-Source Weber Problem with Facility Fixed Cost

Seyed Javad Hosseininezhad[1], Said Salhi[2], Mohammad Saeed Jabalameli[3]

[1] Department of Industrial engineering, K. N. Toosi University of Technology, Tehran, Iran

[2] The Centre for Logistics and Heuristic Optimisation, Kent Business School, University of Kent at Canterbury, Canterbury CT2 7PE, UK

[3] Department of Industrial engineering, Iran University of Science and Technology, Tehran, Iran

Abstract

This paper investigates a capacitated planar location-allocation problem with facility fixed cost. A zone-based fixed cost which consists of production and installation costs is considered. A nonlinear and mixed integer formulation is first presented. A powerful three stage Cross Entropy meta-heuristic with novel density functions is proposed. In the first stage a covering location problem providing a multivariate normal density function for the associated stochastic problem is solved. The allocation values considering a multinomial density function are obtained in the second stage. In the third stage, single facility continuous location problems are solved. Several instances of various sizes are used to assess the performance of the proposed meta-heuristic. Our approach performs well when compared with the optimizer *GAMS* which is used to provide the optimal solution for small size instances and lower/upper bounds for some of the larger ones.

Keywords: *Meta-heuristics, Evolutionary methods, Planar location, Cross Entropy, GAMS.*

1. Introduction

In this study, we are interested in determining the optimal or best location of *m* capacitated facilities in the continuous space with the presence of zone-based facility fixed cost. The objective is to minimize the total cost which includes the cost of transportation, installation, production and the cost for the unmet demand. This new logistical problem is first formulated as a nonlinear and mixed integer programming problem. The software GAMS is used to find optimal solution for small/medium size instance and also to provide lower/upper bounds for the larger ones. A new Cross Entropy meta-heuristic based on three stages is then proposed using appropriate density functions.

We briefly review those recent papers that address planar location problems that are closely related to ours. Since the planar location-allocation (LA) problem, also known as the Multi-Source Weber problem (MSWP), was proposed in the sixties by Cooper (1963), recent advances are put forward including among others Shöbel and Scholz (2010), and Brimberg and Drezner (2013). The former proposed a new approach which they call the Big Cube Small Cube to optimally solve the 2 and 3 facilities in higher dimensions.This is based on the Big Square Small Square method originally given by Hansen *et al.* (1981). The latter developed an improved implementation of Cooper's alternate locate allocate algorithm which is further enhanced by a transfer follow-up. When the facilities have a limited capacity, the problem becomes the capacitated MSWP. Brimberg *et al.* (2014) proposed a new local search approach is embedded within Variable Neighborhood Search for solving the multi-source Weber problem. The algorithm switches between the continuous model and its discrete counterpart until no further improvement can be found in either. Luis *et al.* (2009) provided constructive and adaptive heuristics to generate initial solutions. . Mohammadi *et al.* (2010) used a new method that uses two genetic algorithms for capacitated multi source Weber problem. The first, solves the location problem while the second, solves the allocation problem. A GRASP-based heuristic was also proposed by Luis *et al.* (2011) where adaptive learning is used to construct the restricted candidate list. Akyüz*et al.* (2013) proposed two types of branch and bound algorithms for the capacitated MSWP. The first is an allocation space based-branch and bound algorithm whereas the second is based on the partition of the location space.

Several researches are also in uncertain environment. Durmaz*et al.* (2009) used a discrete approximation technique to address this problem with probabilistic customer locations Mousavi and AkhavanNiaki (2013) studied a capacitated location allocation problem with stochastic location and fuzzy demand. Three fuzzy programming models were developed and a hybrid intelligent algorithm was provided to solve the problem.

The above location allocation problems do not consider fixed cost in the continuous space. However, there are situations in practice which may include zones with high installation cost. These can be modeled using fixed cost. Brimberg *et al.* (2004) introduced the multi-source Weber problem with constant opening cost in the continuous space. They developed a solution method uses a multi-phase heuristic that first solves a discrete version of the problem by existing methods to obtain an estimate of the optimal number of facilities. Brimberg and Salhi (2005) are among the ones that introduced a zone-dependent fixed cost for the incapacitated single facility location-allocation problem in the continuous space. An efficient algorithm that determines the solution optimally was proposed. The authors also provide a simple but an informative illustrative example. For a review on the continuous location problem in general, see Drezner *et al.* (2001), and Brimberg*et al.* (2008).

To the best of our knowledge, this is the first time the meta-heuristic Cross Entropy (CE) is developed to solve this complex continuous location problem.

The remainder of the paper is organized as follows. In the next section, a nonlinear and integer programming formulation is proposed. A meta-heuristic using a Cross Entropy algorithm is presented in section 3. Computational results are given in section 4. Sensitivity Analysis is provided in section 5. The last section summarizes our conclusions and highlights some research areas.

2. A Capacitated Multi-Source Weber Problem with Facility Fixed Cost

Consider there are *n* customers (demand points) indexed by *i* and *m* facilities indexed by *j*. The following notation is used. We first divide the space into *n* zones and introduce a binary variable z_{ji} that defines whether or not facility *j* is located in zone *i*. It is assumed that the distance between each customer and a facility is Euclidean. We formulate the problem as a *0-1* nonlinear and mixed integer programming model. We first provide the necessary notations followed by the mathematical formulation.

Notation

Sets/Indices

N	set of zones (demand points) in the continuous space indexed by i , $\{i=1,2,\ldots,n\}$
K	set of new facilities to be located indexed by j, $\{j=1,2,\ldots,m\}$

Parameters

$A_i = (a_i, b_i)$	Coordinates of customer i
d_i	Demand of customer i
s_j	Capacity of facility j
p_{ji}	Unit production cost of facility j in zone i
f_{ji}	Installation cost of facility j in zone$i$
s_{ji}	Production capacity of facility j in zone i
C_i	Importance of customer i
R	Maximum distance a facility can be located from a center of a zone to be considered as part of that zone
M	Penalty cost for unmet demand which is set to a large amount

Decision variables

$X_j = (x_j, y_j)$	Coordinate of facility j in the plane
$D(X_j,A_i)$	distance between customer i and facility j
T_{ji}	amount supplied from facility j to customer i
z_{ji}	*1* if facility j is located in zone i; *0* otherwise
q_i	Unmet demand of customer i($q_i \le d_i$)

Formulation

$$\min \sum_{i=1}^{n}\sum_{j=1}^{m} T_{ji}\cdot D\left(X_j, A_i\right) + \sum_{j=1}^{m}\sum_{i=1}^{n} z_{ji} p_{ji} \sum_{i=1}^{n} T_{ji} + \sum_{j=1}^{m}\sum_{i=1}^{n} z_{ji} f_{ji} + M \sum_{i=1}^{n} C_i\left(\frac{q_i}{d_i}\right) \tag{1}$$

Subject to

$$\sum_{i=1}^{n} z_{ji}\cdot D(X_j, A_i) \le R, \qquad \forall j = 1,2,\dots,m \tag{2}$$

$$\sum_{i=1}^{n} z_{ji} = 1, \qquad \forall j = 1,2,\dots,m \tag{3}$$

$$\sum_{j=1}^{m} z_{ji} \le 1, \qquad \forall i = 1,2,\dots,n \tag{4}$$

$$\sum_{i=1}^{n} T_{ji} \le \sum_{i=1}^{n} z_{ji}\, s_{ji}, \qquad \forall j = 1,2,\dots,m \tag{5}$$

$$\sum_{j=1}^{m} T_{ji} + q_i = d_i, \;\; i = 1,2,\dots,n \tag{6}$$

$$T_{ji} \ge 0, z_{ji} \in \{0,1\}, q_i \ge 0, X_j = \left(x_j, y_j\right) \in \mathbb{R}^+ \quad \forall i, j$$

Equation (1) is the objective function which is made up of four terms. These include the transportation cost, the production cost, the installation cost and the cost associated with the unmet demand. Constraint set (2) guarantees that if the distance between facility *j* and zone *i* is greater than R, then $z_{ji} = 0$ (i.e., facility *j* does not belong to zone *i*), otherwise $z_{ji} = 1$. Constraint set(3) states that facility *j* is installed only in one zone whereas constraint set (4) shows that only one facility could be located in any zone at most. Constraint set (5) is the capacity constraint (i.e., the production at facility *j* is not violated in zone *i*) and constraint set (6) guarantees that the sum of the assigned and the unmet demand at each customer is equal to the demand of that customer.

This formulation is used in the optimizer *GAMS* to produce the optimal solution for small size problems, and lower and upper bounds for the larger ones. These results will be used to evaluate the performance of a solution method based on the evolutionary meta-heuristic Cross Entropy (CE) which we propose next.

3. A Cross Entropy solution method

In this section we first provide the basic CE algorithm which serves as a basis for our new CE meta-heuristic. This is followed by a subsection explaining some of the main steps of the algorithm.

3.1. The basic CE algorithm

The main idea of Cross Entropy (*CE*) was originally introduced by Rubinstein (1997). CE is an evolutionary technique that is related to the design of an effective learning mechanism which is used throughout the search. For background knowledge, applications and recent references on Evolutionary methods, the reader may consult the interesting paper on location-routing by Prodhon (2011). The main principle of CE is to associate an estimation problem to the original combinatorial optimization problem. This is called the associated stochastic problem which is characterized by a density function ϕ. This stochastic problem is then solved by identifying the optimal sampling density ϕ^*, which is the one that minimizes the Kullback-Leibler distance with respect to the original density ϕ. This distance is also known as the CE between ϕ and ϕ^*. The minimization of the CE leads to the construction of "optimal" updating rules for the density function, and consequently to the generation of improved feasible solutions. The method terminates when the convergence to a point in the feasible region is achieved. The main features of the *CE* algorithm were discussed by de Boer *et al*. (2005).The application of *CE* in the area of optimization can be found in Kroese and Rubinstein (2005). Specific combinatorial problems such as the max-cut problem, the travelling salesman problem and the capacitated vehicle routing problem were solved by Rubinstein and Kroese (2004). A new meta-heuristic scheme for the Integer Knapsack Problem with setups was presented by Caserta *et al.* (2008). The authors proposed a *CE* based scheme with an "intelligent" mechanism aimed at choosing the items to be in the knapsack. Caserta and Quinonez Rico (2009) developed a Lagrangean-based method that transforms the capacitated problem into a set of disjoint uncapacitated lot-sizing problems that are solved by an efficient *CE*-based algorithm. Recently, Bekker and Aldrich (2011) applied *CE* for multi-objective optimization problems. A classical CE algorithm is given in Algorithm 1.

Algorithm 1: A Cross Entropy (CE) for optimization problems	
Step 1:	*Define the initial parameters of the density function ϕ for the associated stochastic problem.*
Step 2:	*Generate a solution vector $(X_1, \ldots, X_N)$ for the problem via Monte Carlo simulation based on the density function ϕ.*
Step 3:	*Calculate the objective function of the problem for each of the solution in $(X_1, \ldots, X_N)$ and select the best solution.*
Step 4:	*Update the parameters of the density function ϕ based on the best solution.*
Step 5:	*If the stopping criterion is satisfied stop, otherwise go to Step 2.*

3. 2.The proposed CE algorithm

In this paper, we present a *CE* algorithm to solve the capacitated multi-source Weber problem defined in Section 2 (Eqs. 1-6). We consider two density functions to generate vectors for identifying the location and the customer allocation for each facility (Step 2). Let us define a family of density functions ϕ on *X*, and use a *2*-dimensional multivariate normal density function for locating facilities under the following probability distribution function:

$$\phi(X,\mu,\Sigma) = e^{-(X-\mu)\Sigma^{-1}(X-\mu)^T/2} \Big/ \frac{1}{\Sigma^{1/2}2\pi} \qquad (7)$$

Where *X*=(*x*,*y*) is *1-by-2* vector representing the *x* and *y* coordinates of facility locations, $\mu = (\mu_x, \mu_y)$ is the mean of facility locations and Σ is a 2-by-2 covariance matrix which is a symmetric positive definite matrix.

We estimate *X* via Monte Carlo simulation. In this regard, *X* could be estimated by drawing a random sample $X^1, \dots, X^N$ from $\phi(X,\mu,\Sigma)$. Hence, the multinomial density function is applied for allocating customers to facilities under the following probability distribution function:

$$\phi(T,s,p) = \frac{s!}{T_1! \cdots T_n!} p_1 \dots p_n \qquad (8)$$

Where s is an integer number, $T = (T_1, \dots, T_n)$ denotes the *1-by-n* vectors of the allocation values and $p = (p_1 \dots p_n)$ is the probabilities of allocating facilities to *n* customers. We wish to estimate *T* via Monte Carlo simulation by drawing a random sample $T^1, \dots, T^N$ from $\phi(T,s,p)$, where *N* is the *CE* population size. At each iteration, the best solutions (elites) found in the previous iterations are also used to update the parameters of the associated probability distribution function. This updating scheme is given next.

The proposed *CE* is made up of three stages. In the first stage, we consider the problem as a continuous covering location problem and find the location of the facilities using (7) until one of the stopping criteria is met. This includes the convergence or the maximum number of iterations whichever comes first. In the second stage, we solve the allocation problem based on the location found in the first stage using (8). Finally, in stage three we solve *m* separate single continuous facility location problems while considering the allocation already found.

3.3. Explanation of the main steps

In *Stage I*, a continuous covering location problem is solved to determine the location of the facilities. In the proposed problem, customer *i* is assigned to facility *j* if $D(X_j, A_i) \le CR$ with CR being a predetermined covering radius. At each iteration the covering matrix is computed. The error value is obtained based on the variance between the best obtained locations for each facility at iteration *t*. The mean of the errors for all facilities is calculated as the error value at iteration *t*. Also, in *Stage II*, the error value is obtained based on the difference between the probability of allocating the demands to facility *j* at iterations *t* and *t-1*.

The mean of the errors for all facilities is then calculated as the error value at iteration *t*. In *Stage III*, a post optimization is carried out based on Cooper's Alternating Transportation-Location Heuristic (Cooper, 1972). Since the allocation values (i.e., the value of q_i) and the cost representing the unmet demand are determined in *phase II*,our problem reduces to solving *m* independent continuous location problems with the aim of minimizing the total cost. We define for each facility *j*, the cost "$FacilityCost_j$" as follows:

$$FacilityCost_j = \sum_{i=1}^{n} T_{ji} \cdot D\left(X_j, A_i\right) + \sum_{i=1}^{n} z_{ji} p_{ji} \sum_{i=1}^{n} T_{ji} + \sum_{i=1}^{n} z_{ji} f_{ji} \; \forall j = 1, .., m \qquad (9)$$

Let $FacilityCost_j^1$ and $FacilityCost_j^2$ be the cost of facility *j* before and after the implementation of *phase III*, respectively. The well known Weiszfeld equations are then used to update the new location of facility *j*:

$$x_j^{(g)} = \frac{\sum_{i=1}^{n} w_{ji} a_i / D(X_j^{(g-1)}, A_i)}{\sum_{i=1}^{n} w_{ji} / D(X_j^{(g-1)}, A_i)}, \quad \text{and} \quad y_j^{(g)} = \frac{\sum_{i=1}^{n} w_{ji} b_i / D(X_j^{(g-1)}, A_i)}{\sum_{i=1}^{n} w_{ji} / D(X_j^{(g-1)}, A_i)},$$

Where $X_j^{(g)} = \left(x_j^{(g)}, y_j^{(g)}\right)$ denotes the location of facility j at iteration g and $A_i = (a_i, b_i)$the coordinates of customer *i*. This procedure is carried out until there is no significant change in the location of facility *j*. The corresponding values of z_{ji} are then determined and the fixed costs including the production cost and the installation cost for facility *j* are computed leading to the evaluation of $FacilityCost_j$ in (9).

In summary, for each facility *j=1,...,m*, if $FacilityCost_j^2 < FacilityCost_j^1$, the new location *j* is selected, otherwise the previous location is retained. A pseudo-code of the proposed *Cross Entropy (CE)* algorithm is given in Algorithm 2.

Algorithm 2: Cross Entropy (CE) for the Capacitated Multi-Source Weber Problem

1: *Determine the population size and the number of elite solutions,* $error^0$ *(initial error), ε(Min error) and maxiter (the max number of iterations),*

Stage I: (Solving the continuous covering problem)

Repeat

2: *Generate the initial multivariate normal distribution parameters (Mean and covariance) based on the x and y coordinates of the zones.*

Draw a sample population $X^1, \dots, X^N \sim MV\ Normal(X, \mu_x^t, \mu_y^t, \Sigma^t)$.

3: ***For*** *each random vector* $X^l = (x_l, y_l)$ *(l=1,…,N), solve the model as a continuous covering location problem*

4: *Select the best solution of the sample population(elites) as* $X_{elite}^t = (x_{elite}^t, y_{elite}^t)$

5: *Update* μ_x, μ_y *and* Σ *as follows:*

$\mu_x^t = mean(x_{elite}^t)$

$\mu_y^t = mean(y_{elite}^t)$

$\Sigma^t = covariance(x_{elite}^t, y_{elite}^t)$

$e_j^t = variance(x_{elite}^t, y_{elite}^t)_j, \qquad \forall\, j = 1, \dots, m$

$error^t = mean(e_j^t)$

Until $(error^t \leq \varepsilon \vee t > maxiter)$ $t \leftarrow t + 1$

Stage II: (Solving the allocation problem)

Repeat

6: *Generate initial multinomial distribution probabilities for allocating demands* (P^0) *and draw a sample population* $T^1, \dots, T^N \sim MNomial(P^t)$.

7: ***For*** *each random vector* T^l *(l=1,…,N) solve the Capacitated Multi-Source Weber Problem.*

8: *Select the best solution for the sample population(elites)*

9: *Update* P^t *as follows* ($T_{ji}^{t,elite}$ *denotes* the best allocation values from facility *j* to customer *i* at iteration *t*):

$$p_j^t = (p_{j1}^t, \dots, p_{jn}^t) = \left(T_{j1}^{t,elite} \Big/ \sum_i T_{j1}^{t,elite}, \dots, T_{jn}^{t,elite} \Big/ \sum_i T_{ji}^{t,elite} \right), \forall\, j = 1, \dots, m$$

$e_j^t = p_j^t - p_j^{t-1} \qquad \forall\, j=1,\dots,m$

$error^t = mean(e_j^t)$

Until $(error^t \leq \varepsilon \vee t > maxiter)$ $t \leftarrow t + 1$

Stage III: (Solving the facility location problem)

10: ***For*** *each facility j (m separate single facility location problem)*

Calculate "$FacilityCost_j^1(X_j^{(Stage\ I)}, T_j^{(Stage\ II)})$"*,where* $X_j^{(Stage\ I)}$ *is the location value obtained in Stage I and* $T_j^{(Stage\ I)}$ *is the allocation values obtained in Stage II for facility j.*

Repeat

Update $X_j^{(g)}$ *based on the Weiszfeld equations (*δ *is Min error):*

Until $(X_j^{(g)} - X_j^{(g-1)} \leq \delta)$ $g \leftarrow g+1$

Calculate "$FacilityCost_j^2(X_j^{(g)}, T_j^{(\boldsymbol{Stage\ II})})$" *value for facility j.*

if $FacilityCost_j^2 < FacilityCost_j^1$ then $X_j^* = X_j^{(g)}$ else $X_j^* = X_j^{(Stage\ I)}$ (where X_j^* is the best location value for *facility j*)

4. Computational results

The *CE* algorithm is implemented in *Matlab* and run on a *Core i5 at 2.53 GHz* with *3GB* of *RAM* memory. The *CE* parameters for *Stage I* and *Stage II* are the following: *(population size, elite size)=(300, 15)* and *(500, 25)*, respectively. The covering radius for stage *I* is *1.5*, *R=0.4 and M=10,000*. We tested our approach in small/medium size instances (*n*=20 to 100 with a step size of 20) and large ones (*n*=200, 300, 400, 500, 1000). The customers are located in the grids of the rectangular areas that depend on the value of *n*. For example for *n*=20, the (*x*-*y*) coordinates of the 20 customers are generated at the grid points of the 5×4 rectangle where customer 1 is at position (1,1), customer 2 at (2,1), etc. The remaining instances are generated in the same way using the following rectangles: 6×5 (*n*=30), 8×5 (*n*=40), 10×10 (*n*=100), 20×10 (*n*=200), 20×15 (*n*=300), 20×20 (*n*=400), 25×20 (*n*=500), 40×25 (*n*=1000). The demand of the customers is randomly generated in the range [1,10]. The capacity of the facilities is based on the size of the problem and it is randomly generated in the range [45, 60]. As we introduced the cost for the unmet demand, we constrained the choice of the capacities in such a way that the sum of the capacities of all the open facilities is smaller than the total customers demand. The fixed cost is randomly generated in the range [800,1000]. The importance of each customer and the unit production cost are both set to 1.

We carried out our experiments using two scenarios. The first one deals with small/medium size instances whereas the second treats the large instances. We compare the obtained *CPU* time and the objective function value of the proposed algorithm with *GAMS* software using *Baron* Solver. We ran the algorithms ten times for each instance and report the following statistics. These include the average (Aver), best value (Best), standard deviation (Std) and the Coefficient of Variation (CV) of the objective function found. The percent deviation from the best-known solutions found by *GAMS* is also reported as (Dev) which is computed as follows:

$$Dev = \frac{F^{best} - F^*}{F^*} \times 100$$

Where F^{best} is the total cost found by the proposed (CE) algorithm and F^* refers to the cost of the best solution (or the optimal/upper bound) found by *GAMS* within the allowed *CPU* time.

Scenario 1 (small/medium size instances)

In this experiment, we report the results for *n*=20 to 100, with *m*= 2 to 10. GAMS is used to yield optimal solutions. According to *Table 1*, the proposed *CE* algorithm provides solutions that are close to those obtained by the optimizer *GAMS*, with deviations being less than 2% in most cases. In addition, our heuristic requires a tiny fraction of the *CPU* time used by GAMS.

Scenario 2 (Large size instances)

We applied our CE algorithm for large instances up to *n=1000* and *m=100, see Table 2*. Though GAMS showed to be relatively slower, we let it run for up to 5 hours so we can report *Lower Bound (LB)* and *Upper Bound (UB)*.The deviation from the UB value and the *CPU* time if the solution is found within the set time of 5hrs are also reported in *Table 2*. In summary, the proposed *CE* algorithm outperformed the optimizer *GAMS* which, in most cases, failed to provide values for *LB* or *UB* within the allowed *CPU* time.

TABLE 1- COMPARISON BETWEEN RESULTS OF THE PROPOSED ALGORITHM (CE) AND GAMS

#	N	m	GAMS (Baron solver)		Proposed Algorithm(CE)								Dev (%)	
					F^{best}				Time(Sec.)					
			Time(Sec)	F^*	Best	Aver	Std	CV	Best	Aver	Std	CV	Best	Aver
1	20	2	0.48	6076	6098	6106	4.83	0.079	6.41	6.64	0.16	2.42	0.36	0.50
2	20	3	0.34	4652	4664	4669	3.20	0.069	9.33	9.95	0.84	8.41	0.27	0.36
3	30	2	0.83	7229	7252	7270	7.68	0.106	7.43	7.71	0.19	2.49	0.33	0.57
4	30	3	0.64	6146	6204	6221	8.33	0.134	10.59	11.35	0.49	4.31	0.94	1.23
5	40	3	0.95	7756	7860	7879	13.81	0.175	11.58	12.39	0.78	6.28	1.34	1.58
6	40	4	1.79	7259	7363	7390	15.97	0.216	15.41	16.22	0.58	3.58	1.43	1.81
7	40	5	2.71	6783	6909	6945	15.58	0.224	19.38	20.82	0.98	4.73	1.86	2.39
8	60	4	4.88	8183	8287	8320	14.30	0.172	31.92	32.65	0.43	1.31	1.26	1.67
9	60	5	38.91	7866	8098	8115	10.54	0.130	38.59	40.64	1.09	2.68	2.95	3.16
10	60	6	55.61	7702	7745	7855	40.36	0.514	43.00	49.23	2.97	6.04	0.56	1.99
11	80	4	8.16	9373	9451	9486	19.03	0.201	37.69	38.91	1.12	2.88	0.84	1.21
12	80	5	14.32	9345	9527	9541	9.08	0.095	45.16	48.28	1.35	2.79	1.94	2.09
13	80	6	90.90	9359	9542	9600	21.80	0.227	56.25	57.33	0.87	1.52	1.95	2.57
14	80	7	354.51	9431	9687	9712	13.99	0.144	62.85	65.21	1.27	1.95	2.71	2.99
15	80	8	376.71	9695	9783	9839	31.73	0.322	75.82	77.64	1.85	2.38	0.91	1.49
16	100	5	28.14	10489	10660	10681	11.61	0.109	52.01	53.71	1.27	2.36	1.63	1.83
17	100	6	97.45	10539	10855	10892	17.66	0.162	63.65	65.02	1.27	1.95	2.99	3.34
18	100	7	264.53	10830	11083	11123	30.93	0.278	73.13	75.34	1.51	2.01	2.34	2.71
19	100	8	408.86	11110	11363	11417	26.22	0.230	79.00	84.34	2.34	2.77	2.28	2.76
20	100	9	477.11	11461	11679	11738	28.64	0.244	91.15	93.68	1.85	1.98	1.90	2.41
21	100	10	1046.08	11752	11993	12071	42.03	0.348	102.51	108.70	3.85	3.54	2.05	2.71

TABLE 2- RESULTS OF THE PROPOSED ALGORITHM (CE) FOR LARGE PROBLEMS

#	N	M	Time(Sec.)				F^{best}				GAMS(Baron solver)			dev. (%)	
			Best	Avr.	Std.	C.V.	Best	Avr.	Std.	C.V.	Time(Sec.)	LB	UB	Best	Avr.
1	200	10	134.68	141.23	4.72	3.34	17217	17259	24	0.14	995	15800	17279	**-0.36**	**-0.12**
2	200	15	230.52	233.26	2.38	1.02	21110	21146	29	0.14	5007	18964	20751	1.73	1.90
3	200	20	301.71	306.22	3.60	1.17	25657	25770	57	0.22	18000	22479	25349	1.22	1.66
4	300	10	169.11	176.04	5.64	3.20	18550	18572	13	0.07	4829	17050	18385	0.90	1.02
5	300	15	294.55	313.45	0.00	0.00	22884	22910	18	0.08	15085	20800	22562	1.43	1.54
6	300	20	386.85	391.24	3.86	0.99	27388	27483	62	0.23	18000	NF	NF	-	-
7	300	30	563.12	582.62	10.26	1.76	37810	38054	128	0.34	18000	NF	NF	-	-
8	400	10	224.36	230.39	4.76	2.07	19218	19241	18	0.10	7148	17675	19291	**-0.37**	**-0.26**
9	400	20	486.10	498.20	9.17	1.84	28571	28647	51	0.18	18000	NF	NF	-	-
10	400	30	718.84	732.34	12.03	1.64	38667	38893	157	0.40	18000	NF	NF	-	-
11	400	40	950.97	973.66	12.56	1.29	50516	50755	213	0.42	18000	NF	NF	-	-
12	500	10	265.01	276.11	7.72	2.80	19456	19481	23	0.12	18000	17500	19549	**-0.48**	**-0.35**
13	500	20	591.29	610.39	11.10	1.82	29204	29310	77	0.26	18000	NF	NF	-	-
14	500	30	859.47	879.22	14.81	1.68	39596	39780	127	0.32	18000	NF	NF	-	-
15	500	40	1163.28	1185.85	18.21	1.54	51290	51622	188	0.36	18000	NF	NF	-	-
16	500	50	1423.45	1474.23	32.35	2.19	64494	64914	248	0.38	18000	NF	NF	-	-
17	1000	10	562.31	586.63	15.47	2.64	20330	20352	18	0.09	18000	NF	NF	-	-
18	1000	20	1276.87	1309.92	22.59	1.72	30889	30979	67	0.22	18000	NF	NF	-	-
19	1000	30	1892.03	1916.06	21.10	1.10	42033	42216	109	0.26	18000	NF	NF	-	-
20	1000	40	2485.21	2533.18	29.40	1.16	53678	53941	186	0.34	18000	NF	NF	-	-
21	1000	50	3085.14	3131.95	23.37	0.75	66141	66384	191	0.29	18000	NF	NF	-	-
22	1000	100	7151.88	7218.83	46.86	0.65	138602	139247	657	0.47	18000	NF	NF	-	-

NF= No feasible solution Found

According to the results from both tables it can be noted that there are very small variations in the results of the proposed *CE* algorithm which demonstrates the robustness of our method. This claim is also shown the *CV* values. It can also be observed that the post optimization stage was influential as it improves the solution with an average of about 1% and 2% for small/medium and large size instances, respectively while requiring a negligible extra CPU time.

For illustration purposes, we consider *Stage II* to show the overall reduction in the objective function values in terms of the number of iterations. *Figures 1A* and *1B* present the pattern with *n=500* and *m=40*, and *n=1000* and *m=100*, respectively. Similar patterns are also observed for most instances. In summary, the number of iterations required to find the solution is recorded as 50 for $m \leq 10$, 60 for $10 < m \leq 50$ and 70 for $m > 50$.

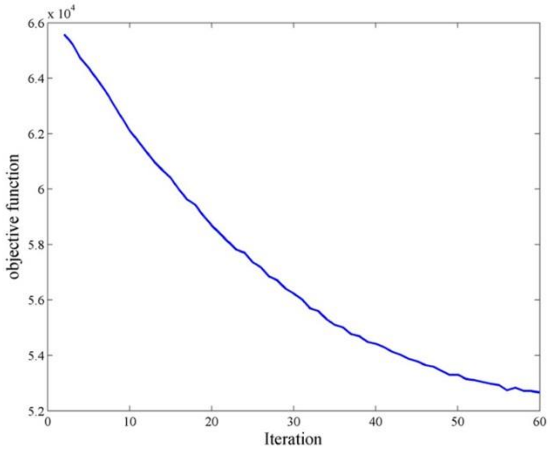

FIGURE 1 (A) OBJECTIVE FUNCTION VS. ITERATION NUMBER (N = 500, M = 40) FOR STAGE II.

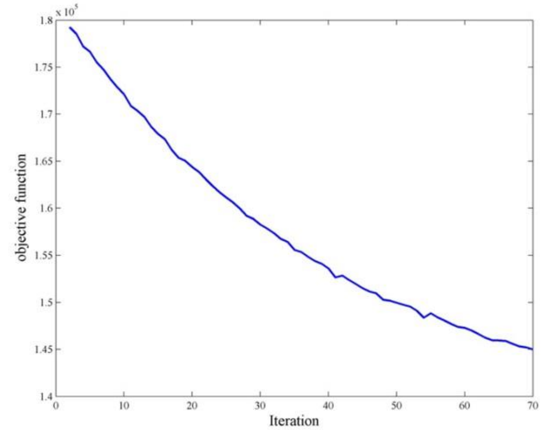

FIGURE 1 (B) OBJECTIVE FUNCTION VS. ITERATION NUMBER (N = 1000, M = 100) FOR STAGE II.

5. Conclusion

In this section, we analyze sensitivity of two parameters of the model (CR as a predetermined covering radius and R which is maximum distance a facility can be located from a center of a zone to be considered as part of that zone) based on the average solution (Av.Dev.), best solution (BestDev.) and average time solution (Av.Time). Also, we recorded the results for a given instance with $n=100$ and $m=10$ and used a combination for CR in [1, 2] and R in [0.2, 0.6]. The results are also reported in *Table 3.*

TABLE 3- SENSITIVITY ANALYSIS OF THE PROPOSED MODEL (WITH N=100 AND M=10)

#	CR	R	GAMS (Baron solver)		Proposed Algorithm(CE)								Dev (%)	
					F^{best}				Time(Sec.)					
			Time(Sec)	F^*	Best	Aver	Std	CV	Best	Aver	Std	CV	Best	Aver
1	1	0.20	1721.30	11924	12076	12154	37.43	0.308	105.89	122.43	13.84	11.30	1.28	1.96
2	1	0.40	1046.08	11752	12099	12143	26.27	0.216	109.36	122.97	8.49	6.91	2.95	3.33
3	1	0.60	1168.92	11635	12131	12163	26.65	0.219	108.75	118.28	4.86	4.11	4.26	4.54
4	1.5	0.20	1721.30	11924	12048	12124	53.38	0.440	108.41	111.15	1.86	1.68	1.04	1.68
5	1.5	0.40	1046.08	11752	11992.7	12071	42.03	0.348	102.51	108.70	3.85	3.54	2.05	2.71
6	1.5	0.60	1168.92	11635	11993.1	12117	65.25	0.539	105.68	109.98	2.80	2.54	3.08	4.14
7	2	0.20	1721.30	11924	12065	12149	56.49	0.465	113.21	116.96	2.78	2.38	1.18	1.88
8	2	0.40	1046.08	11752	12030.8	12117	49.21	0.406	111.28	118.66	5.10	4.29	2.37	3.10
9	2	0.60	1168.92	11635	12096.2	12136	24.79	0.204	111.35	115.15	2.22	1.93	3.97	4.31

Figures 2A and *2B* present the sensitivity analysis of BestDev. and Av.Dev. values based on R amounts, respectively. As shown, by increasing the *R*, the BestDev. and Av.Dev. values are increased. It seems to CR=1.5 results in better solutions which we considered for our instances.

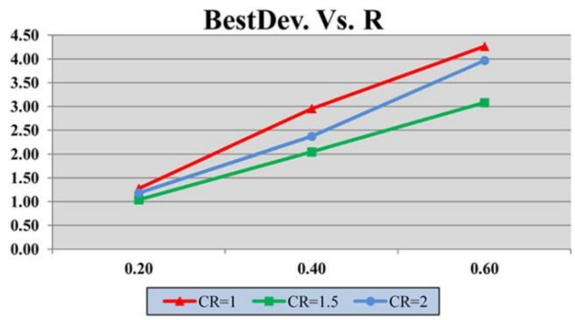

FIGURE 2 (A) BEST DEV. VS. R. VS. R.

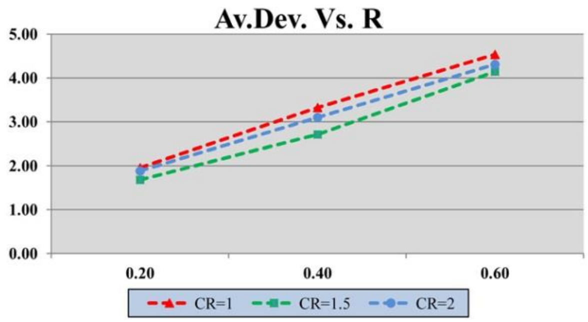

FIGURE 2 **(B)** AVERAGE DEV. VS. R

Figures 3A and *3B* present the sensitivity analysis of BestDev. and Av.Dev. values based on CR amounts, respectively. As shown, by increasing the *CR=1.5* and *2* results in better solution in the BestDev. and Av.Dev. values. Therefore, based on the above analysis, *R* is more sensitive than *CR* in providing better solution for the proposed model.

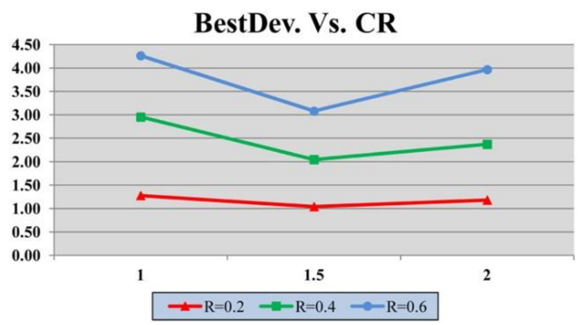

FIGURE 3 (A) BEST DEV. VS. CR.

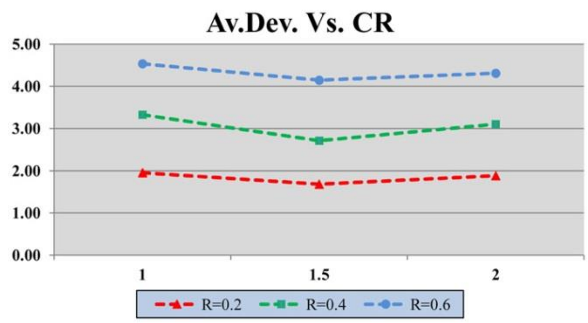

FIGURE 3 **(B)** AVERAGE DEV. VS. CR.

Figures 4A and *4B* present the sensitivity analysis of Av.Time. values based on *R* and *CR* amounts, respectively. As shown, *CR* is more sensitive than *R* in CPU time for solving the proposed model.

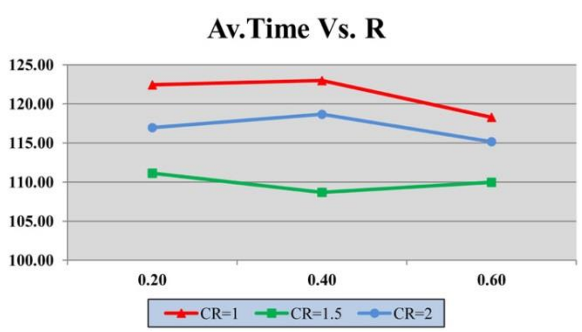

FIGURE 4 (A) AVERAGE TIME VS. R.

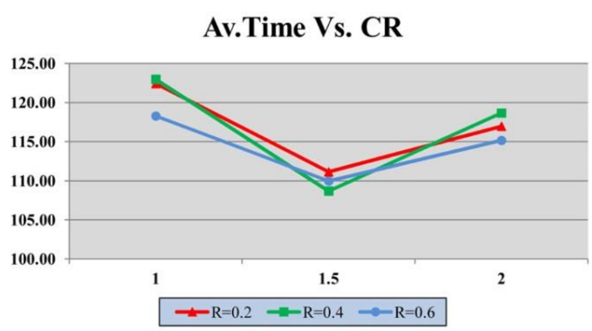

FIGURE 4 **(B)** AVERAGE TIME VS. CR.

6. Conclusion

A capacitated multi-source Weber problem with zone-based fixed cost is investigated. A nonlinear mixed integer programming model is presented and tested on small/medium size instances using *GAMS*.

An algorithm based on *Cross Entropy (CE)* is also proposed using appropriate multivariate normal and multinomial density function for the associated stochastic problem. The proposed algorithm generated competitive results producing a deviation of less than 2% in most small/medium size instances while requiring a tiny fraction of the time needed by GAMS. In particular, for the large size instances, *GAMS* fails to generate lower and upper bounds in most cases even after 5 hours whereas our CE heuristic maintains its performance throughout the experiments.

The present work can be extended to include an adaptive and powerful search approach. This can be achieved by adopting a reinforced learning which continually feeds the information found in *Stage III* back into *Stage I.* The present CE meta-heuristic could also be adapted to address other related continuous location problems by appropriately defining the corresponding density functions.

7. References

Akyüz M. H., Altınel I. K., Öncan T. (2013), "Location and allocation based branch and bound algorithms for the capacitated multi-facility Weber problem", *Annals of Operations Research*, (in Press).

Bekker J., Aldrich C., (2011), "The cross-entropy method in multi-objective optimization", *European Journal of Operational Research*, 211, 112–121.

Brimberg J., Drezner Z., (2013), "A new heuristic for solving the p-median problem in the plane", *Computers & Operations Research*, 40, 427-437.

Brimberg, J., Drezner, Z., Mladenović, N., and Salhi, S., (2014), “A New Local Search for Continuous Location Problems”, *European Journal of Operational Research* 232, 256-265.

Brimberg J., Hansen P., Mladenovic D., Salhi S., (2008), "A Survey of Solution Methods for the Continuous Location-Allocation Problem", *International journal of Operation Research*, 5,1-12.

Brimberg J., Salhi S., (2005), "A Continuous Location-Allocation Problem with Zone-Dependent Fixed Cost", *Annals of Operations Research*, 136, 99–115.

Brimberg, J., Mladenović, N. and Salhi, S., (2004), “The Multi-Source Weber Problem with Constant Opening Cost”, *Journal of the Operational Research Society* 55, 640-646.

Caserta M., Quinonez Rico E., (2009), "A cross entropy-Lagrangean hybrid algorithm for the multi-item capacitated lot-sizing problem with setup times", *Computers & Operations Research*, 36, 530 – 548.

Caserta M., Quinonez Rico E., Marquez Uribe A.,(2008), "A cross entropy algorithm for the Knapsack problem with setups", *Computers & Operations Research*, 35, 241 – 252.

Cooper L., (1963), "Location-Allocation problems", *Operations Research*, 11, 3, 331–343.

Cooper L., (1972), “The transportation-location problem”, *Operations Research*, 20, 94–108.

de Boer P., Kroese D.P., Mannor S., Rubinstein R.Y., (2005), "A tutorial on the cross-entropy method", *Annals of Operations Research*, 134, 19–67.

Drezner Z., Klamroth K., Schöbel A., Wesolowsky G., (2001), The Weber problem. In: Drezner Z, Hamacher HW (eds) Facility location: applications and theory. Springer, Berlin.

DurmazE., ArasN.,Altınel I. K., (2009), "Discrete approximation heuristics for the capacitated continuous location–allocation problem with probabilistic customer locations", *Computers & Operations Research*, 36, 7, 2139–2148.

Hansen P., J-F. Thisse., (1981), “ The generalized Weber-Eawls problem”. In *Operational Research ’81* (Hamburg, 1981), pp 569-577, North Holland, Amsterdam.

Kroese D.P., Rubinstein R.Y., (2005), "The cross-entropy method for combinatorial optimization, rare event simulation and neural computation", *Annals of Operations Research*, 134 (1).

Luis M., Salhi S., Nagy G., (2011), "A guided reactive GRASP for the capacitated multi-source Weber problem", *Computers & Operations Research*, 38, 1014–1024.

Luis M., Salhi S., Nagy G., (2009), "Region-rejection based heuristics for the capacitated multi-source Weber problem", *Computers & Operations Research*, 36, 2007 – 2017.

Mohammadi, N., Malek, M. R., and Alesheikh, A. A., (2010), “A New GA Based Solution for Capacitated Multi Source Weber Problem”, *International Journal of Computational Intelligence Systems* 3, 514-521.

Mousavi S. M., AkhavanNiaki S. T., (2013), "Capacitated location allocation problem with stochastic location and fuzzy demand: A hybrid algorithm", *Applied Mathematical Modelling*, (in Press).

Prodhon C., (2011), "A hybrid evolutionary algorithm for the periodic location-routing problem", *European Journal of Operational Research*, 21, 204–212.

Rubinstein R. Y., (1997), "Optimization of computer simulation models with rare events", *European Journal of Operational Research*, 99, 89-112.

Rubinstein R.Y., Kroese D.P., (2004), “The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization”, Monte-Carlo Simulation, and Machine Learning. Springer.

Shöbel A., Scholz S., (2010), “The big cube small cube solution method for multidimensional facility location problems*”, Computers & Operations Research*, 37, 115–122.