10-2016

Simulation and Optimization Of Ant Colony Optimization Algorithm For The Stochiastic Uncapacitated Location-Allocation Problem

Jean-Paul Arnaout

Georges Arnaout
Old Dominion University

John El Khoury

SIMULATION AND OPTIMIZATION OF ANT COLONY OPTIMIZATION ALGORITHM FOR THE STOCHASTIC UNCAPACITATED LOCATION-ALLOCATION PROBLEM

Jean-Paul Arnaout*

Business Administration Department
Gulf University for Science and Technology
Kuwait

Georges Arnaout

Department of Engineering Management and Systems Engineering
Old Dominion University
Norfolk, VA

John El Khoury
Department of Civil Engineering
Lebanese American University
Byblos, Lebanon

(Communicated by Mutsunori Yagiura)

Abstract. This study proposes a novel methodology towards using ant colony optimization (ACO) with stochastic demand. In particular, an optimization-simulation-optimization approach is used to solve the Stochastic uncapacitated location-allocation problem with an unknown number of facilities, and an objective of minimizing the fixed and transportation costs. ACO is modeled using discrete event simulation to capture the randomness of customers' demand, and its objective is to optimize the costs. On the other hand, the simulated ACO's parameters are also optimized to guarantee superior solutions. This approach's performance is evaluated by comparing its solutions to the ones obtained using deterministic data. The results show that simulation was able to identify better facility allocations where the deterministic solutions would have been inadequate due to the real randomness of customers' demands.

1. **Introduction and literature review.** In the continuous uncapacitated location allocation problem (*also referred to as multi-facility weber problem (MWP)*), the aim is to locate m new facilities that will serve n known demand points with an objective of minimizing the fixed costs of opening facilities and variable costs of transportation. The literature reports that most of the models developed for the facility location problem are very hard to solve to optimality and classified as NP-hard [15].

Due to the computational complexity of the problem, the literature offers limited exact methods for its solution. Earlier studies were able to solve for 2 facilities and 30 customers using branch and bound [10]. The problem size increased to cover 2

to 100 facilities and 287 customers with Krau [9] that used a combination of B&B, global optimization, and Column generation.

Since exact solutions were not appropriate for practical scenarios where the problem size is usually large, researchers have focused more on heuristics and meta-heuristics for the deterministic version of the problem. Bischoff and Klamroth [5] presented several heuristics to solve the capacitated version of the problem with rectilinear distances. Aras et al. [2] tackled the problem by partitioning the feasible region into a finite set of domains and solving the corresponding mixed-integer subproblems. Bischoff et al. [4] provided two heuristics for the MWP with barriers, and reported that their algorithms can attain solutions of reasonably sized multi-facility locationallocation problems with barriers, both with regard to computation time and solution quality. Brimberg et al. [6] compared between Genetic Algorithms (GA), TS, and Variable Neighborhoud Search (VNS), noted that the heuristics' solutions deteriorate when the number of facilities increases, with VNS performing better. Arnaout [3] introduced an Ant Colony Optimization (ACO) algorithm for the same problem with deterministic demand. ACO was compared to GA, VNS, TS, and the previous' superiority proven. For more on deterministic location-allocation literature, the reader can refer to Jabalameli and Ghaderi [8], Aras et al. [2], Salhi and Gamal [18], Gamal and Salhi [7], Liu and Xu [11], and Pasandideh and Niaki [17].

As shown above, most of the literature has focused on the deterministic version of the problem. Unfortunately, and in realistic scenarios, customers demand are always changing; i.e. customers have a stochastic demand and not a deterministic one. Logendran and Terrell [12] were the first to introduce the stochastic uncapacitated version and they modeled price sensitive stochastic demands with an objective of maximizing expected profits. Zhou [21] tackled the same problem with stochastic demand using an expected value model, chance-constrained programming and dependent-chance programming. In a later study, Zhou and Liu [20] generated new stochastic models for the capacitated variant of the problem. Finally, Mehdizadeh et al. [13] introduced a new hybrid algorithm for the capacitated version also.

More recent studies by Ozkisacik et al. [16] and Altinel et al. [1] addressed the probabilistic version of the problem, where they considered the customer locations to be randomly distributed. Their work differed from this research as the latter considers random customer demand but known locations.

In this study, we modify the ACO that was introduced for the deterministic problem in [3] to account for the stochastic nature of the problem, where customers demands follow uniform distributions. Following this, the stochastic ACO (referred to hereafter as ACO_S) is modeled using discrete event simulation, and then the algorithm's parameters are optimized using a combination of metaheuristic procedures in order to reach superior solutions. In other words, our novel approach optimizes the simulated ACO_S parameters in order to optimize the solution of the problem at hand.

To the best of the author's knowledge, there does not exist published research that addresses the problem with unknown m and stochastic demand, nor research that models ant colony using discrete event simulation and uses optimization with simulation to determine the algorithm's parameters.

2. ACO_S **application to MWP.** The objective function minimizes the *TotalCosts* (transportation and fixed costs) and is formulated as:

$$Min \sum_{i=1}^{n} \sum_{j=1}^{m} D_{ij}.Z_{ij}.T.d(i,j) + \sum_{j=1}^{m} k_j.F \tag{1}$$

where, n is the number of demand sites, $d(i,j)$ is the distance between demand point i and facility j, k_j is the index of new facilities ($k_j = 1$ if facility j is open, 0 otherwise), and $m = n$ is a suitable upper bound on the number of facilities as highlighted in [3]. Necessary parameters and variables are defined as follows:

- x_{F_j}, y_{F_j}: decision variables that indicate respectively the x and y coordinates of new facility j.
- x_{D_i}, y_{D_i}: respectively the x and y coordinates of demand point i.
- F: fixed cost of opening a new facility.
- T: transportation cost per unit of distance per unit demand.
- D_i: stochastic demand / number of trips to demand point i.

As discussed earlier, the ACO_S will be modeled using simulation. Furthermore, the MWP with m unknown can be solved in three ACO_S stages: first, we determine the number of facilities m; second, we decide on initial temporary estimates for the facilities' coordinates (x_{F_j}, y_{F_j}); and third, we determine the assignment of demand points n to the facilities m based on the latter's estimated coordinates, and we compute the final exact coordinates of each facility. The algorithm simulation logic is depicted in Figure 1, where at the initialization phase an entity is created to assign appropriate values to the pheromone trails of each of the three stages: $\tau_j^I, \tau_{ij}^{II}, \tau_{ij}^{III}$.

2.1. **Stage 1: Determining the number of facilities m.** The first stage consists of deciding on the value of m, the number of facilities to open. We represent the first stage with a vector (S_1) that contains n entries representing the number of facilities m and each entry is populated with a facility number ($j = 1, , m = n$). For instance, if we have 10 customer points ($n = 10$), then the following vector: $S1 = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$ will represent the possible edges that an ant could move to. If an ant moves to edge [3] in (S_1), it means that $m = 3$ facilities in this ant's tour. The Pheromone trail (τ_j^I) is defined for this stage to indicate the favorability of choosing the number of facilities from (S_1). In addition to the pheromone, we assist the algorithm with (η_j^I) to solve the problem. The probability to select m is calculated as shown in (2) and (η_j^I) is calculated using (3), which suggests the greedy heuristic of minimizing the number of facilities in order to minimize the fixed costs.

$$\Pi_j^I = \frac{(\tau_j^I)^\alpha.(\eta_j^I)^\beta}{\sum_{l \in \Psi}(\tau_l^I)^\alpha.(\eta_l^I)^\beta} \tag{2}$$

$$\eta_j^I = \frac{1}{j} \tag{3}$$

Following Equations (2) and (3) and as highlighted in Figure 1, the cumulative distribution function $(CDF_I(j))$ is populated. Next, a random variable (rv) between 0 and 1 is generated based on which m is determined. Using this probabilistic approach in choosing m (*versus solving for all its possible values*) is better in terms of computational time, especially as m increases.
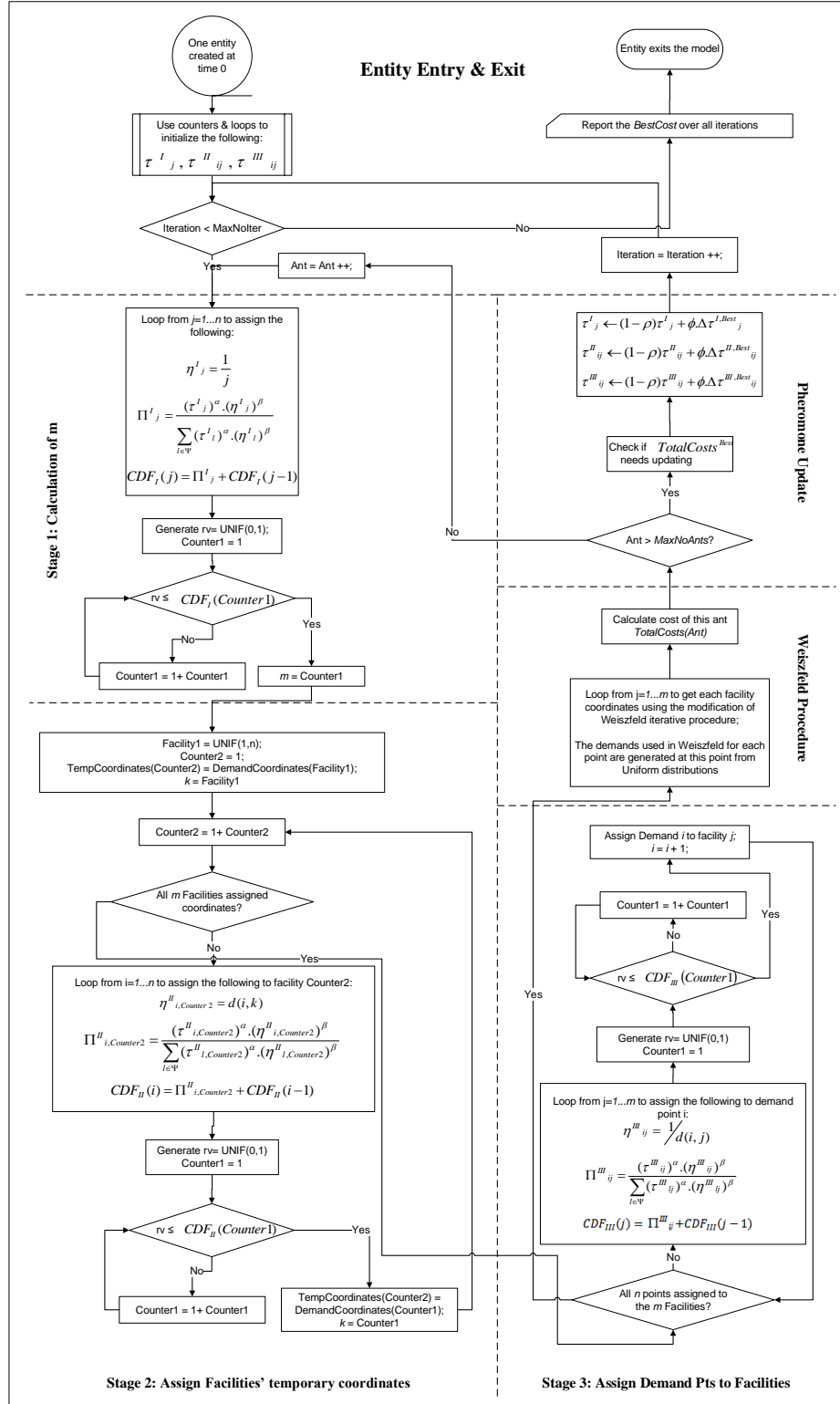
FIGURE 1. Simulation Model Logic

2.2. **Stage 2: Initial temporary facilities' coordinates.** In Stage 2, initial temporary estimates for the facilities' coordinates are assigned. Note that the m facilities should be located far from each other. Having two facilities next to each other is inefficient because they are uncapacitated; i.e. it makes more sense to replace them with one single facility. The coordinates' estimates are obtained by putting them equal to the ones of m demand points, where the latter are chosen as follows. The first facility ($j = 1$) coordinates are set to be equal to the ones of a randomly selected demand point. Now that one facility has coordinates (x_{F_1}, y_{F_1}), (η_{ij}^{II}) for the remaining facilities are computed following (4), where $d(i, k)$ is the Euclidean distance between demand point i and facility k, with k referring to the predecessor facility (*with an already assigned temporary coordinates*). The probability to set the coordinates of facility j to be equal to the one of demand point i is calculated as shown in (5).

$$\eta_{ij}^{II} = d(i, k) \tag{4}$$

$$\Pi_{ij}^{II} = \frac{(\tau_{ij}^{II})^\alpha . (\eta_{ij}^{II})^\beta}{\sum_{l \in \Psi}(\tau_{lj}^{II})^\alpha . (\eta_{lj}^{II})^\beta} \tag{5}$$

The Pheromone trail (τ_{ij}^{II}) is defined for this stage to indicate the favorability of choosing the coordinates of demand point i for facility j's temporary coordinates. Similar to Stage 1, $CDF_{II}(i)$ is populated and rv is generated, based on which each facility is assigned temporary coordinates. The second stage output can be represented by a vector (S_2) that contains m entries representing the demand points from which we obtained facility j's temporary coordinates. For example, if we determined $m = 4$ from Stage 1, then S_2=[2 10 4 6] indicates that the first facility's coordinates were set to be equal to the coordinates of the second demand point ($i = 2$), the second facility to $i = 10$'s coordinates, the third to $i = 4$, and the forth facility to $i = 6$'s coordinates.

2.3. **Stage 3: Allocation of demand points to facilities.** Now that the m facilities have initial coordinates, we determine in Stage 3 the assignment of demand points n to the facilities. A Pheromone trail (τ_{ij}^{III}) is defined for this stage to indicate the favorability of assigning demand point i to facility j. The probability to assign i to j is calculated as shown in (6) and (η_{ij}^{III}) is calculated using (7), which suggests the greedy heuristic of minimizing the distances between the demand points and the facilities.

$$\Pi_{ij}^{III} = \frac{(\tau_{ij}^{III})^\alpha . (\eta_{ij}^{III})^\beta}{\sum_{l \in \Psi}(\tau_{lj}^{III})^\alpha . (\eta_{lj}^{III})^\beta} \tag{6}$$

$$\eta_{ij}^{III} = \frac{1}{d(i, j)} \tag{7}$$

The third stage output can be represented by a vector (S_3) that contains n entries representing the demand points. Each entry is populated with the assigned facility number ($j = 1, , m$). Note that this assignment was based on the temporary facilities' coordinates from Stage 2, and it will be used to determine the actual facilities' coordinates as highlighted in Section 2.4.

2.4. **Facilities coordinates, total costs, and pheromone update.** After the third stage, each facility has a group of towns assigned to it. We use here a modification of the iterative procedure proposed by Weiszfeld [19] to come up with the facilities' coordinates. For every facility j, the steps below are implemented, where n_j refers to the demand points that are assigned to facility j.

Step 1. Set the inital values of x_{F_j} and y_{F_j} according to 8:

$$x_{F_j} = \frac{\sum_{i=1}^{n_j} x_{D_i}}{n_j}, y_{F_j} = \frac{\sum_{i=1}^{n_j} y_{D_i}}{n_j} \tag{8}$$

Step 2. For $i = 1, ..., n_j$,
  {if($x_{F_j} = x_{D_i}$ and $y_{F_j} = y_{D_i}) \rightarrow (x' = x_{D_i}; y' = y_{D_i})$}
  else, exit for-loop and go to **Step 4.**
Step 3. For each demand point i, evaluate x' and y' as defined in 9:

$$x' = \frac{\sum_{i=1}^{n_j} \frac{D_i T x_{D_i}}{d(i,j)}}{\sum_{i=1}^{n_j} \frac{D_i T}{d(i,j)}} and, y' = \frac{\sum_{i=1}^{n_j} \frac{D_i T y_{D_i}}{d(i,j)}}{\sum_{i=1}^{n_j} \frac{D_i T}{d(i,j)}} \tag{9}$$

Step 4. If($x' = x_{F_j}$ and $y' = y_{F_j}$), STOP
  Else, set($x_{F_j} = x'$ and $y_{F_j} = y'$) and go to **Step 2.**
Step 5. $(x_{F_j}, y_{F_j})=(x', y')$

After all ants finish their paths, we update the pheromone amounts in each link locally by reducing the amounts due to evaporation. We also update the pheromone amounts in each link globally by increasing the ones constructed by the ant that produced the least $TotalCosts$. This is estimated according to 10, 11, and 12.

$$\tau_j^I \longleftarrow (1 - \rho)\tau_j^I + \phi.\Delta\tau_j^{I,Best} \tag{10}$$

$$\tau_{ij}^{II} \longleftarrow (1 - \rho)\tau_{ij}^{II} + \phi.\Delta\tau_{ij}^{II,Best} \tag{11}$$

$$\tau_{ij}^{III} \longleftarrow (1 - \rho)\tau_{ij}^{III} + \phi.\Delta\tau_{ij}^{III,Best} \tag{12}$$

Where:

$$\Delta\tau_j^{I,Best} = \begin{cases} \frac{1}{TotalCosts^{Best}}, & \textbf{if } j \textbf{ is used by best ant} \\ 0, & \textbf{Otherwise} \end{cases}$$

$$\Delta\tau_{ij}^{II,Best}; \Delta\tau_{ij}^{III,Best} = \begin{cases} \frac{1}{TotalCosts^{Best}}, & \textbf{if arc}(i,j) \textbf{ is used by best ant} \\ 0, & \textbf{Otherwise} \end{cases}$$

The innovative aspect about our solution methodology is the use of three successive trails to solve the problem: one for determining the number of facilities, another one for generating the facilities' temporary coordinates, and one more for assigning the demand points to the facilities. Note that D_i follows uniform distribution as described in Section 4.

3. **Optimization of ACO_S parameters.** As can be seen from Section 2, ACO_S has several parameters that will regulate the algorithm's performance. In particular, before simulating ACO_S, we need to find the optimal values for the following parameters (along with their ranges): $Ants : (5, 60)$; $\rho : (0.01, 0.3)$; $\varphi : (0.01, 0.3)$; $\alpha, \beta : (1, 2)$; $\tau_j^I, \tau_{ij}^{II}, \tau_{ij}^{III} : (1, 10)$. The ranges were adopted from Arnaout [3]. In order to decide on the optimal ACO_S parameters, an optimization tool is used with simulation as shown in Figure 2. In particular, $OptQuest$ from $OpTek$ Systems, Inc. is a computer software system that allows users to automatically search for optimal solutions to complex systems, and is integrated within $Arena$, the Simulation software that was used in this study. The optimization software based on its integrated heuristics, decides on values for the ACO_S parameters. The latter is used in the simulation run and the $TotalCosts$ are recorded. $TotalCosts$ is input back into the optimization tool to decide on appropriate values of ACO_S parameters for the next simulation. This loop is repeated for as many replications and simulations needed
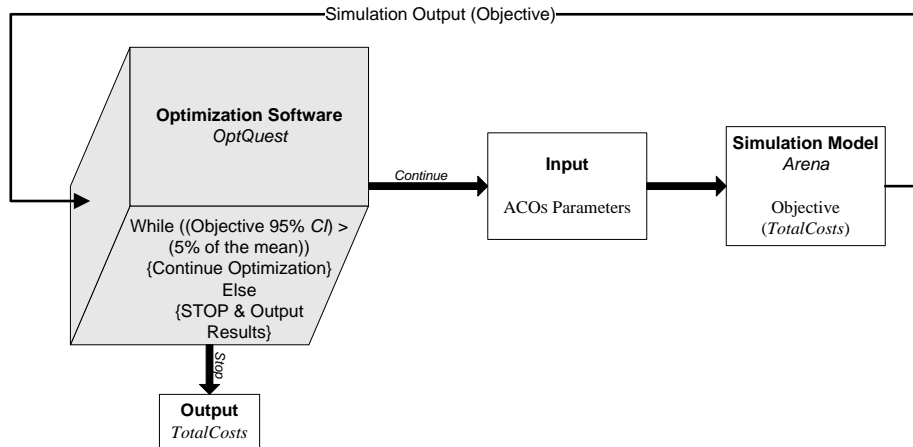
FIGURE 2. Optimization Approach for ACO_S

to guarantee a minimum $TotalCosts$ with a 95% Confidence Interval that is at most 5% from the average $TotalCosts$.

4. **Computational tests.** The proposed ACO_S was modeled on *Arena* 13.0 Simulation Software from Rockwell Systems, running on Windows XP with a Pentium 4 processor at 2.33 GHz and 2 GB of RAM. The algorithms were run on small (5 and 20 stochastic demand points) and large (60 and 100 stochastic demand points) problems and each size was tested with 5 instances of the problem. The reason stochastic data was used for the customers' demand is to ensure a more realistic representation of the supply chain environment. The demands are stochastic following uniform distributions: $D_i = U[0.8D_{d_i}, 1.2D_{d_i}]$, where D_{d_i} is the demand of customer i in the deterministic case as generated in Arnaout [3]. In particular, the data used for the demand (D_i) and coordinates of the demand points came from randomly generated values from a uniform distribution $U[1, 100]$. The reason uniform distributions were used is due to their high variances, ensuring that the presented heuristics are being tested under unfavorable conditions. Furthermore, the algorithms were compared under different dominance of F and T as follows. The selection of the **fixed** cost of opening a facility (F) and the transportation (**variable**) cost (T) values determines the level of dominance. That is, when the fixed and variable costs are balanced (denoted by F, T *Balanced*), they are set to $F = \$30,000$ and $T = \$30$. When fixed costs are dominant (denoted by F *Dominant*), then the values of F and T are $\$30,000$ and $\$10$ respectively. When the transportation costs are dominant (denoted as T *Dominant*), then F and T are set to $\$30,000$ and $\$50$ respectively. Following this, a total of 60 problem instances were solved by each algorithm. This dominance approach was used to cover the different scenarios that might exist in practice, where the ratio of fixed and variable costs is highly dependent on the industry type and its location. Furthermore, the values of F and T were adopted from Arnaout [3] to have a fair comparison between ACO_S and ACO_D.

4.1. **Deterministic ACO (ACO_D).** As one of the objectives of this study is to highlight the advantage of simulating stochastic demand versus using directly the deterministic ACO, the latter was implemented as follows.

The MWP with stochastic demand can be solved using the deterministic ACO (ACO_D) in Arnaout [3] that was modeled using Visual C++ and can only handle deterministic data. This is done by using the average of the demands' stochastic distributions (i.e. D_{d_i}) as input, running ACO_D to output the number of facilities, their locations and allocations. Next, these outputs are simulated under the stochastic demand in order to generate the actual $TotalCosts$.

In Arnaout [3], a design of experiments was utilized to determine the most suitable ACO_D parameters that will minimize $TotalCosts$. These parameters were used in this study and they are as follows: $Ants = 60$; $\rho = 0.01$; $\varphi = 0.01$; $\alpha = 1$; $\beta = 1$; $\tau_j^I = 2.555$; $\tau_{ij}^{II} = 1$; $\tau_{ij}^{III} = 10$ and $MaxNoIter = 10363$.

4.2. **Results.** Table 1 and Figure 3 show the average and maximum $TotalCosts$ for all problem structures and all algorithms. In particular, Table 1 shows the average and maximum $TotalCosts$ attained, along with the number of new facilities recommended by both ACO_D and ACO_S. It can be seen from the latter that the number of facilities is different when the simulation was used to model ACO and capture the stochastic demand, leading to lower $TotalCosts$. It can be seen that ACO_S performed better than ACO_D in all the 60 instances. This is due to the fact that a more suitable number and allocation of facilities was attained after simulating the randomness of the demand.

TABLE 1. Avg and Max $TotalCosts$; Avg and Max δ for all Problems

	Demand	ACO_D			ACO_S			δ	
		Facilities	Avg	Max	Facilities	Avg	Max	Avg	Max
F Dominant	5	3	91418	93468	2	77430	77658	18	20
	20	4	245229	248614	3	220412	220894	11	13
	60	7	627527	644356	6	502022	503403	25	28
	100	10	945643	971370	8	716396	724903	32	34
F, T Balanced	5	4	120563	123091	3	99451	99746	21	23
	20	8	388249	400121	10	363961	367935	7	9
	60	18	1108033	1138302	23	872467	27882405	27	29
	100	29	1833518	1884079	38	1378585	1385353	33	36
T Dominant	5	4	127561	131693	4	123327	123328	3	7
	20	10	460391	472180	10	427723	434819	8	9
	60	26	1459744	1499851	33	1186784	1199881	23	25
	100	42	2534624	2604796	55	1964825	1973330	29	32

Even though ACO_S outperformed ACO_D in all problem instances, and as the latter's solutions appeared to be close to ACO_S (see Figure 3), the percentage deviation (δ) of ACO_D from ACO_S was used as a measure of performance for each problem instance. That is:

$$\delta = \frac{TotalCosts_{ACO_D} - TotalCosts_{ACO_S}}{TotalCosts_{ACO_S}} \times 100\% \tag{13}$$

Figure 4 shows the average values of δ for the three dominance combinations for all problems. It can be clearly seen that ACO_D's deviation from ACO_S increased with the problem size. Another important note about the δ values is that the average and maximum deviations of ACO_D for all problem instances were 20% and 36% respectively.

4.3. **Computational times.** As highlighted earlier, ACO_D was modeled using Visual C++ for deterministic demand, while ACO_S was modeled using discrete event simulation in order to account for the stochastic demand. Consequently,
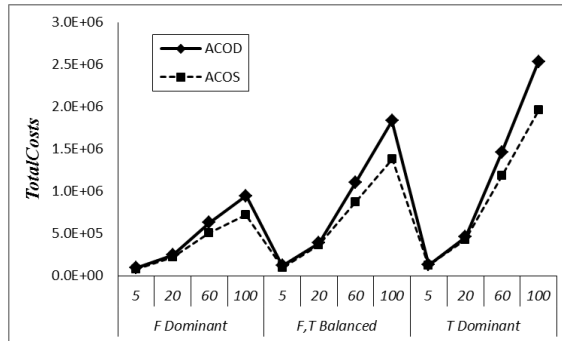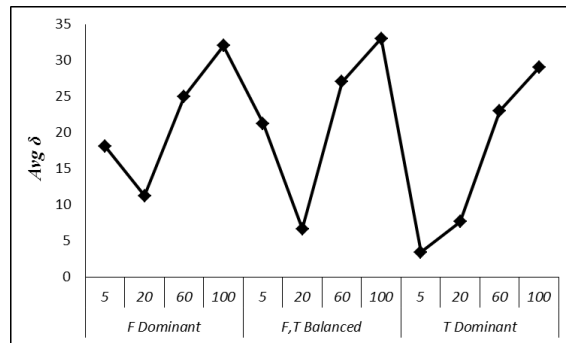
FIGURE 3. Average $TotalCosts$ for all problems

FIGURE 4. Average δ for all problems

comparing the computational times for the two ACO approaches would not be a reasonable assessment. Furthermore, the solution of ACO_D is simulated in order to generate the $TotalCosts$, and ACO_S's parameters are optimized to generate the $TotalCosts$; i.e., both will require an additional computational time beyond the algorithms' scope.

Having said this, and to give the reader a sense of the computational performance, Table 2 depicts the average times (minutes) for the algorithms (excluding ACO_D's simulation and ACO_S's optimization). As can be seen, both ACO_D and ACO_S have similar performance; this is logical because both follow similar algorithmic steps. Furthermore, it can be noted from Arnaout [3] that ACO_D requires less time to converge to more superior solutions than Genetic Algorithms (GA); subsequently, the same would hold for ACO_S.

5. **Conclusions.** While lots of research has tackled the deterministic continuous facility location-allocation problem, few works addressed the stochastic version of the problem, and no previous published studies were found on the problem with unknown number of facilities with stochastic demand, nor research that modeled ACO using discrete event simulation and used optimization with simulation to determine the ACO parameters.

TABLE 2. Computational Times

		Small Problems		Large Problems	
	Demand Points	5	20	60	100
ACO_S	F Dominant	3.62	13.83	47.77	79.86
	F, T Balanced	1.03	16.63	64.23	123.39
	T Dominant	0.92	10.6	74.42	180.58
ACO_D	F Dominant	3.53	14.08	47.57	77.42
	F, T Balanced6	1.04	16.74	62.51	122.16
	T Dominant	0.89	10.98	74.19	188.95

In this paper, we have modeled a previously introduced ACO ([3]) using discrete event simulation to account for the randomness of customers' demands for the Stochastic Euclidean facility location-allocation problem with unknown number of facilities. The differences between this study and its predecessor ([3]) are as follows:

1. In Arnaout [3], ACO was modeled using Visual C++. On the other hand, in this study ACO was modeled using simulation. Up to the author's knowledge, no literature exists that models Ant Colony Optimization using simulation; in particular, discrete event simulation. This allowed for the introduction of stochastic demand in this study.

2. The computational results indicated a significant reduction in costs when ACO was simulated in comparison to its previous deterministic implementation in Arnaout [3]. In particular, a reduction of up to 36% in total costs was attained. Furthermore, better facilities allocations were achieved when the randomness of data was taken into account.

3. One of the challenges of ACO is finding suitable parameters. In Arnaout [3], design of experiments was used to find suitable parameters. In this study, an optimization approach was used with Simulation to find the optimal ACO parameters. This by itself is another addition to the body of knowledge. In particular, optimization decides on values for the ACO_S parameters. The latter is used in the simulation run and the $TotalCosts$ are recorded. $TotalCosts$ is input back into the optimization tool to decide on appropriate values of ACO_S parameters for the next simulation. This loop is repeated for as many replications and simulations needed in order to guarantee a minimum $TotalCosts$. In other words, we are optimizing an optimization method (ACO_S) using Simulation.

REFERENCES

[1] I. K. Altinel, K. C. Ozkisacik and N. Aras, Variable neighborhood search heuristics for the probabilistic multi-source weber problem, *Journal of the Operational Research Society*, **62** (2011), 1813–1826.

[2] N. Aras, M. Orbay and I. K. Altinel, Efficient heuristics for the rectilinear distance capacitated multi-facility Weber problem, *Journal of the Operational Research Society*, **59** (2008), 64–79.

[3] J-P. Arnaout, Ant Colony Optimization algorithm for the Euclidean location-allocation problem with unknown number of facilities, *Journal of Intelligent Manufacturing*, **24** (2013), 45–54.

[4] M. Bischoff , T. Fleischmann and K. Klamroth, The multi-facility location-allocation problem with polyhedral barriers, *Computers and Operations Research*, **36** (2009), 1376–1392.

[5] M. Bischoff and K. Klamroth, An efficient solution method for Weber problems with barriers based on genetic algorithms, *European Journal of Operational Research*, **177** (2007), 22–41.

[6] J. Brimberg, P. Hansen, N. Mladenovi and E. Taillard, Improvements and comparison of heuristics for solving the uncapacitated multisource weber problem, *Operations Research*, **48** (2000), 444–460.

[7] M. D. H. Gamal and S. Salhi, Constructive heuristics for the uncapacitated location-allocation problem, *Journal of the Operational Research Society*, **52** (2001), 821–829.

[8] M. Jabalameli and A. Ghaderi, Hybrid algorithms for the uncapacitated continuous location-allocation problem, *International Journal of Advanced Manufacturing Technology*, **37** (2008), 202–209.

[9] S. Krau, *Extensions du Problème de Weber*, Ph.D thesis, Ecole Polytechnique de Montreal, 1996.

[10] R. Kuenne and R. M. Soland, Exact and approximate solutions to the multisource Weber problem, *Mathematical Programming*, **3** (1972), 193–209.

[11] W. Liu and J. Xu, A study on facility location-allocation problem in mixed environment of randomness and fuzziness, *Journal of Intelligent Manufacturing*, **22** (2011), 389–398.

[12] R. Logendran and M. P. Terrell, Uncapacitated plant location-allocation problems with price sensitive stochasticdemands, *Computers and Operations Research*, **15** (1988), 189–198.

[13] E. Mehdizadeh, M. Tavarroth and S. Nousavi, Solving the Stochastic Capacitated Location-Allocation Problem by Using a New Hybrid Algorithm, *Proceedings of the 15th WSEAS International Conference on Applied Mathematics*, (2010), 27–32.

[14] M. Ohlemuller, Tabu search for large location-allocation problems, *Journal of the Operational Research Society*, **48** (1997), 745–750.

[15] S. H. Owen and M. S. Daskin, Strategic facility location: A review, *European Journal of Operational Research*, **111** (1998), 423–447.

[16] K. C. Ozkisacik, I. K. Altinel and N. Aras, Solving probabilistic multi-facility Weber problem by vector quantization, *OR Spectrum*, **31** (2009), 533–554.

[17] S. Pasandideh and S. Niaki, Genetic application in a facility location problem with random demand within queuing framework, *Journal of Intelligent Manufacturing*, (2010).

[18] S. Salhi and M. D. H. Gamal, A genetic algorithm based approach for the uncapacitated continuous location-allocation problem, *Annals of Operations Research*, **123** (2003), 203–222.

[19] E. Weiszfeld, Sur le point par lequel la somme des distances de n Points donnés est Minimum, *Tohoku Mathematical Journal*, **43** (1937), 355–386.

[20] J. Zhou and B. Liu, New stochastic models for capacitated location-allocation problem, *Computers and Industrial Engineering*, **45** (2003), 111–125.

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

E-mail address: arnaout.j@gust.edu.kw

E-mail address: georges.arnaout@gmail.com

E-mail address: john.khoury@lau.edu.lb