A COMPARISON OF ALGORITHMS FOR LOGISTICS-ORIENTED VEHICLE ROUTING PROBLEMS

John E. Bell, Department of Operational Sciences, Air Force Institute of Technology, 2950 Hobson Way, WPAFB Ohio 45433, john.bell@afit.edu
Stanley E. Griffis, Department of Operational Sciences, Air Force Institute of Technology, 2950 Hobson Way, WPAFB Ohio 45433, stanley.griffis@afit.edu

ABSTRACT

This research compares the metaheuristic method of Ant Colony Optimization (ACO) to the Clark-Wright Savings algorithm on a set of logistics-oriented vehicle routing problems (VRP). ACO simulates the decision-making processes of ant colonies as they forage for food and is similar to Tabu Search, Simulated Annealing and Genetic Algorithms. Experimentation shows that ACO is successful in finding solutions near the best-known solutions for small problems with twenty demand locations. In addition, the type of spatial pattern does appear to make a difference in the ability of the two algorithms to find improved solutions. ACO is superior to the savings algorithm for this problem set and it needs to be tested on larger logistics-oriented VRP.

INTRODUCTION

The vehicle routing problem is an important logistics problem that has been extensively studied for several decades. The traditional single depot vehicle routing problem includes simultaneously determining the routes for several vehicles from a central supply depot to a number of customers and returning to the depot without exceeding the capacity of each vehicle. This problem is of economic importance due to the costs associated with providing and operating delivery vehicles to transport products to a set of geographically dispersed customers. When an organization is able to reduce the length of its routes or decrease the number of its vehicles, it is able to provide better service to its customers and potentially obtain a more profitable and competitive position. The problem typically involves minimizing costs of the combined routes for a number of vehicles and quite often distance traveled by the vehicles is used as a surrogate for the objective of cost. In the past, the majority of research on the vehicle routing problem has used demand sets with randomly generated customer locations. However, it has been recognized by logistics researchers that patterns exist in the spatial dispersion of customers. This research uses problems, which represent five different spatial patterns believed to actually occur in the logistics distribution of products in real-world markets.

The process of selecting a route for an individual vehicle allows the selection of any combination of customers. Therefore, the vehicle routing problem is considered a combinatorial optimization problem where the number of feasible solutions for the problem increases exponentially with the number of customers. Additionally, the vehicle routing problem is related to the traveling salesman problem where an out and back tour from a central location is determined for each vehicle. Since there is no known algorithm that will find the optimal solution in every instance, the vehicle routing problem is considered NP-hard [16]. For such problems, the use of heuristics is considered a reasonable approach in finding solutions to the vehicle routing problem. This paper compares the ability of a well-known traditional heuristic, the Clark-Wright Savings algorithm, to the modern metaheuristic technique of ant colony optimization (ACO).

The Savings algorithm is a construction type algorithm, which adds locations to each vehicle route based on the associated potential savings of including the location. This deterministic method has the ability to quickly determine a solution and will always find the same solution to a problem. In contrast, ACO simulates the behavior of ant colonies as they find the most efficient routes from their nests to food sources. The decision making process of ants is embedded in the search algorithm of a group of virtual ants which are then used to find improved solutions to the vehicle routing problem. This method is more time exhaustive, but has the ability to generate a variety of solutions for each problem. The comparison of the two algorithms is relevant because the Savings algorithm was found to be the algorithm of choice when logistics-oriented vehicle routing problems were first introduced, and ACO is one of a class of algorithms that have not been tested on such problems. The results indicate that the performance of VCO may be superior for generating solutions to logistics-oriented VRP.

VEHICLE ROUTING PROBLEM

The vehicle routing problem has been an important problem in the field of logistics for at least the last four decades [11]. It is described as finding the minimum distance or cost of the combined routes of a number of vehicles *m* that service a number of customers *n*. This system is mathematically described as a weighted graph G= (V,A, d) where the vertices are represented by V = $\{ v_0, v_1, \ldots v_n \}$, and the arcs are given by A=$\{(v_i, v_j)$: $i \neq j\}$. A central depot where each vehicle starts its route is located at v_0, and each of the other vertices represents one of the *n* customers. The distances associated with each arc are represented by the variable d_{ij} and each demand location is assigned a non-negative demand q_i. Additionally, each vehicle is given a capacity constraint, *Q*. The problem is subject to the following constraints: each customer is serviced by only one vehicle, each vehicle must start and end its route at the depot, v_0, and the total demand serviced by a vehicle cannot exceed *Q*. Finally, the VRP in this study is symmetrical and $d_{ij} = d_{ji}$ for all *i* and *j*.

The vehicle routing problems analyzed in this paper have been created to simulate the dispersal of demand patterns, which actually may occur in logistics distribution, and are therefore considered "logistics-oriented". The problems studied here were created by Agarwal [1] and studied extensively by logistics researchers in the early 1990s [2][3]. Five classes of problems are included in this set and each class includes ten randomly generated problems. The problem classes are random, cluster, sector, urban-rural, and coastal. The first problem class, random, is similar to other uniformly distributed problem sets on a Euclidean plane and is comparable to commonly tested VRP in operations research [9][10]. The cluster class includes problems where a number of stops are grouped together in clusters that are separated spatially from other clusters. The sector class is a pie-shaped demand area resulting from dividing a 360 degree area into several sections, as might assigned to separate distribution agents or vehicles. The urban-rural class includes a large number of demand locations gathered around the centrally located depot, and also includes a group of scattered demand locations farther away from the depot/city that represent rural demand. The Coastal or "West Coast" class represents the linear distribution of demand locations created by geographic structures such an ocean or mountain ranges. Illustrations of these classes are depicted in [2].

Research has been accomplished on the vehicle routing problem [9][10] using advanced meta-heuristic approaches such as Tabu Search [15] and Simulated Annealing [18], and ant colony optimization [7][8][5]. This research applies a version of ant colony optimization, which uses candidate lists to generate solutions to a set of logistics-oriented VRP not analyzed previously by any of the modern metaheuristic class of algorithms. The solutions generated by ACO are compared to the solutions generated by the Clark-Wright Savings algorithm in previous research [2][3].

SAVINGS ALGORITHM

The savings algorithm used here is the method first described by Clark and Wright [11]. It starts by inefficiently assuming that a single truck services each demand location. Then, in a constructive manner, routes are eliminated by combining locations and routes that offer the greatest distance savings potential. In this manner, vehicle routes are simultaneously determined as route numbers are decreased. This process continues until vehicle capacities are met or no additional savings can be achieved. As pointed out by [10], the savings algorithm can be designed as either a sequential or parallel algorithm. In the sequential version, a single route is constructed until the vehicle capacity is met, and in the parallel version, more than one vehicle route is simultaneously constructed. This paper uses a sequential version of the savings algorithm. It is unknown which version was used in [2][3].

ANT COLONY OPTIMIZATION

Ant colony optimization (ACO) is a metaheuristic technique that uses artificial ants to find solutions to combinatorial optimization problems such as the VRP. ACO simulates the abilities of real ants and additionally possesses enhanced features such as memory of previous actions and knowledge about the distances to other demand locations. Ant colonies have the ability to solve complex problems and successfully find and collect food through the use of a chemical substance called pheromone. As an individual ant travels between its nest and a food source, it deposits amounts of pheromone proportional to the quality of the food source in order to communicate preferred paths to other ants. Each ant may move somewhat randomly, however, the probability of an ant selecting a particular path increases with the amount of pheromone on the path. The random selection of paths by individual ants also leads to the discovery of alternate routes and insures navigation around obstacles. Because it takes less time for ants to traverse a short path, the accumulation of pheromone is greater, thereby increasing the chance that other ants will similarly follow the shorter path. In addition, since pheromone slowly evaporates over time, the probability that ants will use less desirable routes also decreases. The process of route selection by ants can be described as pseudo-random proportional process [13] and is a primary element of ACO. More detailed descriptions of ant behavior as it relates to ACO are provided by [14].

ACO FOR VEHICLE ROUTING

The use of ant colonies was first applied to the traveling salesman problem and the quadratic assignment problem [12] and has since been applied to other problems such as the space planning problem [6], the machine tool tardiness problem [4] and the multiple objective JIT sequencing problem [17]. Additionally, ant colony optimization was first applied to the VRP by [7]. Since then, a ranked ACO approach was given by [8], and a multiple colony approach was offered by [5]. Although, many techniques and variations of ant colony optimization exist, the general features of the algorithm typically include methods for route construction, pheromone trail updating and final route improvement.

Route Construction

With ACO, an individual artificial ant simulates a vehicle, and routes are constructed by allowing each ant to select customers until all customers are visited. Initially, each ant starts at the depot and the set of customers included in the tour is empty. The ant selects the next customer from the list of feasible locations and the vehicle's storage capacity is updated before another customer location is selected. When the capacity constraint of a vehicle is met, the ant returns to the depot. After all of the customers are visited, the ant returns to the depot, and the total distance *L* is computed for the complete route of the artificial ant. Using this technique, complete individual routes are generated sequentially by a predetermined number of ants *m*. Additionally, each ant must construct a vehicle route that visits each customer. To select the next customer location *j*, the ant uses the following formula [10]:

$$j = \arg\max \left\{ (\tau_{iu})(\eta_{iu})^{\beta} \right\} \quad \text{for } u \notin M_k, \quad \text{if } q \le q_0; \text{ otherwise S} \tag{1}$$

where τ_{iu} is equal to the amount of pheromone on the path between the current location *i* and possible locations *u*. The value η_{iu} is the inverse of the distance between the two customer locations, and the parameter β establishes the importance of distance in comparison to pheromone quantity in the selection algorithm. The ants working memory, M_k keeps track of locations already visited by an ant and which are no longer considered for selection. The value q is a random uniform variable [0,1] and the value q_0 is a parameter. When an ant selects a new location to visit, the arc with the highest value from equation (1) is selected unless q is greater than q_0. In that case, the ant selects a random variable (S) to be the next location to visit based on the probability distribution of p_{ij}, which favors high levels of pheromone and short paths:

$$p_{ij} = \frac{(\tau_{ij})(\eta_{ij})^{\beta}}{\sum_{u \notin M_k} (\tau_{iu})(\eta_{iu})^{\beta}} \qquad \text{if } j \notin M_k \text{ otherwise } 0 \tag{2}$$

Using formulas (1) and (2) each ant either follows the most favorable path, or randomly selects a path to follow based on the probability distribution established by distance and pheromone accumulation. This selection process continues until each customer is visited and the tour is complete.

Pheromone Trail Updating

The pheromone trails of ants must be updated to reflect the colonies performance and the quality of the solutions found. Trail updating includes local updating of trails after individual solutions have been generated and global updating of the route for the best solution found after a predetermined number of solutions *m* has been generated. Local updating is conducted by decreasing the amount of pheromone on all visited arcs in order to mimic the natural evaporation of pheromone and to ensure that no one path becomes too dominant. This is done with the following local trail updating equation,

$$\tau_{ij} = (1-\alpha)\tau_{ij} + (\alpha)\tau_0 \tag{3}$$

where α is a parameter that controls evaporation speed and τ_0 is set to an initial pheromone value assigned to all arcs in the graph G. For this study, τ_0, has been set to the inverse of the previously best-known route distance for each problem, a technique used in previous research [9].

After a predetermined number of complete routes have been constructed, global trail updating is performed by adding pheromone to each of the arcs included in the best route found by one of *m* ants. Global trail updating is accomplished according to the following relationship,

$$\tau_{ij} = (1-\alpha)\tau_{ij} + \alpha(L)^{-1} \tag{4}$$

This equation encourages the use of short routes and increases the probability that new routes will use the arcs contained in previous best solutions. This process is repeated for a predetermined number of iterations and the overall best solution from all of the iterations is presented as an output of the model. It is believed that this final solution should represent a good approximation of the optimal solution.

Final Route Improvement

The route construction and pheromone updating processes described above are typical for ACO as it is applied to the traveling salesman problem [10]. However, research by [4] shows that the attainment of improved solutions to the VRP is dependent on route improvement strategies in the algorithm. The first strategy involves the inclusion of a local exchange procedure to act as an improvement heuristic within the routes found by individual vehicles. This technique tests pairwise exchanges of customer locations visited by individual vehicles to see if an overall improvement in the objective function can be attained by changing the order in which customers are visited. If any of these solutions is found to improve the objective function, then the best solution is modified prior to conducting pheromone updating for the route. This process adds to the number of individual combinations explored by the search and can be thought of as solving several Traveling Salesman Problems after assigning the customers to vehicles [5].

The second improvement strategy is the use of a candidate list for determining the next location selected in a vehicle route. Each individual location is assigned a candidate list based on the distance to all other locations in the location set. Only the closest locations are included in the candidate list for the current location and are made available for selection as the next location to be visited in the route. The best size of the candidate list has been determined to approximately ten to twenty locations by previous research [5][10]. This research uses a candidate list size of ten for all ACO searches. It is believed that this restriction prevents the algorithm from wasting time considering locations that are a great distance from the current location and have very little chance of creating an improved solution.

EXPERIMENTATION

The savings algorithm and ant colony optimization algorithm are tested on the fifty logistics-oriented vehicle routing problems [1] in order to compare the ability of the algorithms to find improved solutions to problems with varying spatial characteristics. The results are also compared to the results of previous research that used the savings algorithm and to the best known solutions for each problem.

Design of Experiment

A description of the fifty problems used in the experiment is presented in Table 1. The location coordinates for all locations and the depot are available in [1]. All problems have a vehicle capacity of one hundred units and each problem consists of twenty stops. The best-known solutions were believed to be optimal and were generated by [1] using a set-partitioning approach.

Solutions for each problem have been generated using ACO thirty times in order to understand the central tendency and variances associated with the results of the ant colony optimization search for improved solutions and to make meaningful statistical comparisons. The savings algorithm generated the same solution in each recomputation of the algorithm; therefore multiple runs were not performed for this algorithm. The measures of comparing performance include the mean route distance *L,* minimum route distance *L,* and the percentage inferiority of the minimum route distance *L* in comparison to the best known solutions to the problem.

Generation of all Savings and ACO solutions for each problem was done using C++ coding on an Athlon AMD4 900 MHz processor. For all ACO solutions, search parameters that were found to be robust in previous research were used: $\alpha = .1$, $\beta = 2.3$, $q_0 = .9$, and m=25. Each run of the ACO model consisted of 5000 iterations of the trail construction and trail updating processes.

TABLE 1: Problem Characteristics and Previous Solution Values

Class	Problem	n	Q	best known	Class	Problem	n	Q	best known
Random	1	20	100	444	Cluster	1	20	100	790
	2	20	100	457		2	20	100	718
	3	20	100	455		3	20	100	770
	4	20	100	513		4	20	100	828
	5	20	100	501		5	20	100	776
	6	20	100	567		6	20	100	773
	7	20	100	530		7	20	100	814
	8	20	100	537		8	20	100	820
	9	20	100	542		9	20	100	741
	10	20	100	476		10	20	100	767
Urban	1	20	100	392	Coast	1	20	100	423
	2	20	100	414		2	20	100	403
	3	20	100	410		3	20	100	460
	4	20	100	413		4	20	100	390
	5	20	100	411		5	20	100	403
	6	20	100	440		6	20	100	444
	7	20	100	389		7	20	100	394
	8	20	100	460		8	20	100	358
	9	20	100	391		9	20	100	415
	10	20	100	449		10	20	100	385
Sector	1	20	100	502	Sector	6	20	100	519
	2	20	100	583		7	20	100	552
	3	20	100	543		8	20	100	517
	4	20	100	558		9	20	100	542
	5	20	100	446		10	20	100	487

EXPERIMENTAL RESULTS

The results of the experiment listed in Table 2, reveal that the ACO approach is able to generate improved solutions for the VRP in comparison to the savings algorithm solutions for the randomly generated problem class. In three of the ten problems, the best solution found by the ant colony algorithm was actually smaller than the previously best-known solution. In addition, the best ACO solutions were no more than 1.62% larger than the best known solutions and six of the ten best solutions were either the same or better than the known optimal. Additionally, in five of the ten problems, all thirty runs found the best solution. This lack of variability indicates the consistent ability of the ACO algorithm to generate near optimal solutions in comparison to the Savings algorithm, which had solutions ranging from 13% to 39% larger than the previously best-known solution to the problem.

TABLE 2: Results - Random Class

Problem	1	2	3	4	5	6	7	8	9	10
Initial	1307.11	1494.4	1378.66	1541.82	1631.95	1624.13	1616.94	1575.08	1461.66	1555.38
Best Known	444.14	457.38	454.88	513.08	501.37	566.77	529.64	536.57	541.85	475.45
Savings	629.48	616.89	544.24	581.03	698.31	654.23	602.44	728	753.02	585
Ant Best	**443.26**	**456.8**	454.88	**477.8**	501.37	566.77	538.22	537.93	548.32	475.45
Ant Best %	-0.20%	-0.13%	0.00%	-6.88%	0.00%	0.00%	1.62%	0.25%	1.19%	0.00%
Ant Ave	459.383	456.8	454.88	509.8663	502.638	573.476	538.22	537.93	550.1173	475.45

The results for the cluster class in Table 3 were consistent with those for the random class. Again, the ACO algorithm was able to find improved solutions compared to the previously best-known solutions for three of the ten problems. Additionally, the best solution found by the ant colony algorithm was equal to or better than the previously know best solution in 7 out of 10 problems. Also, the variation of the results for the ACO algorithm was impressive. In four of ten problems, the ant algorithm found its best solution in all thirty trials. In all ten problems, the ACO algorithm's best solution was no more than .73% above the value of the previously best known solution, compared to the savings algorithm results which ranged from .5% to 24% larger than the previously best known solutions.

TABLE 3: Results - Cluster Class

Problem	1	2	3	4	5	6	7	8	9	10
Initial	2008.59	1913.65	1969.19	2119.16	2014.16	1981.62	2010.42	2096.57	1927.34	1892.07
Best Known	790.49	718.31	769.86	827.97	776.48	773.49	814.46	819.94	741.74	767.05
Savings	827.22	871.87	784.12	832.78	797.88	819.61	892.63	865.6	751.6	951.97
Ant Best	790.49	718.53	771.81	827.97	**742.81**	773.49	814.46	825.92	**700.27**	**766.94**
Ant Best %	0.00%	0.03%	0.25%	0.00%	-4.34%	0.00%	0.00%	0.73%	-5.59%	-0.01%
Ant Ave	790.49	726.164	776.1633	827.97	771.1673	774.104	814.46	826.0267	735.4147	766.94

The results for the sector class differ from previous results for both the ACO algorithm and the savings algorithm. First, the savings algorithm actually performed well and found three solutions slightly improved compared to previously known best solutions. The ant colony algorithm dominated this problem set and its best solution was better than any previously best-known solution in 8 out of 10 problems. Only in problem four was the savings algorithm able to find the best solution. In 9 out of 10 problems the ACO algorithm was able to find improved solutions compared to previously best solutions and the differences ranged from –11.29% to 3.48%. In problem 2, the savings algorithm was still unable to find a near optimal solution and was 29% inferior to the previously best-known solution.

TABLE 4: RESULTS - SECTOR CLASS

Problem	1	2	3	4	5	6	7	8	9	10
Initial	2105.05	2675.71	2157.86	1973.61	1654.14	2214.99	2199.54	2072.54	2435.5	1685.89
Best Known	502.18	582.94	542.59	557.87	446.04	519.03	552.48	517.23	541.78	487.37
Savings	**494.9**	751.53	609.73	**538.58**	502.87	653.75	579.66	555.69	625.37	**471.45**
Ant Best	**445.49**	**540.34**	**515.56**	**543.04**	**419.71**	**513.77**	571.73	**511.31**	**538.7**	**456.81**
Ant Best %	-11.29%	-7.31%	-4.98%	-2.66%	-5.90%	-1.01%	3.48%	-1.14%	-0.57%	-6.27%
Ant Ave	456.4593	597.0433	542.1413	546.7597	442.8083	518.8547	574.7813	519.361	585.0097	460.0843

The results for the urban-rural class of problems listed in Table 5 were not as successful for either of the algorithms. Only in problem 4 was the ACO algorithm able to find an improved solution in compared to the previously known best. The best solutions for the ant colony algorithm ranged up to 9.28% larger than the known best solution, and only in one problem did the ACO algorithm find its best solution in all thirty of the trials. Additionally, the savings algorithm found difficulty with this problem type and its results ranged from 14.3% to 63.9% larger than the best-known solutions for the problem.

TABLE 5: RESULTS – URBAN RURAL CLASS

Problem	1	2	3	4	5	6	7	8	9	10
Initial	1037.85	1047.49	1165.83	1107.43	1020.12	1092.25	1244.94	1157.28	1049.39	1137.7
Best Known	391.52	413.75	410.07	412.54	411.31	440.29	389.29	459.54	391.32	448.78
Savings	551.31	580.39	468.71	492.23	493.75	584.68	560.5	532.49	641.5	561.14
Ant Best	401.24	419.97	414.87	**380.13**	422	474.81	408.31	502.17	405.22	463.18
Ant Best %	2.48%	1.50%	1.17%	-7.86%	2.60%	7.84%	4.89%	9.28%	3.55%	3.21%
Ant Ave	411.3107	433.936	427.8877	415.207	432.0063	489.95	408.31	516.2353	424.3023	464.161

During examination of the data for the final problem class, it was discovered that the third problem was not listed in [1] and was not reported in the results of [3]. Therefore, the results in Table 6 only include the nine problems which were analyzed. The results for this problem were not as dramatic as previous problems. ACO was only able to improve on the best-known solution in one of the ten problems and this improvement was only –1.95%. The solutions for the remaining nine problems ranged from 0.00% to 3.6% inferior to previously best-known solutions. The savings algorithm also was unable to improve on the best-known solutions and results ranged from 2.0 to 28.3% greater than the best-known solutions.

TABLE 6: RESULTS - COASTAL CLASS

Problem	1	2	3	4	5	6	7	8	9	10
Initial	1421.5	1288.51	-	1160.13	1280.59	1294.33	1542.35	1180.88	1239.54	1106.02
Best Known	423.23	401.41	460	390.48	403.2	443.57	393.97	357.69	414.59	384.87
Savings	431.85	449.49	-	437.62	523.61	569.01	518.47	456.56	482.08	486.49
Ant Best	**414.99**	403.94	-	403.6	407.88	450.25	393.97	357.93	429.5	395.43
Ant Best %	-1.95%	0.63%	-	3.36%	1.16%	1.51%	0.00%	0.07%	3.60%	2.74%
Ant Ave	433.278	406.2413		405.5953	408.416	450.25	393.97	359.613	429.5	400.69

DISCUSSION AND CONCLUDING REMARKS

Ant colony optimization shows advantages in a head to head comparison against the Clark-Wright Savings algorithm for five different classes of problems. In 17 out of 49 problems capable of being analyzed, the ACO algorithm was able to find an improved solution to the best known solutions to the twenty location logistics-oriented VRP previously published. The savings algorithm used in this research was only able to obtain a new best solution superior to one found by ACO in one of the Sector problems. In addition, the variation of solution quality was much higher for the savings algorithm and in many cases the ACO algorithm found its best solution in all thirty trials for the particular problem. In no case did the ACO solution differ by more than 9.28% from the previously best-known solution. In contrast, the savings algorithm values were as much as 63.9% greater than the best solution.

The ability of the algorithms to solve spatially different problems also varied. The ACO algorithm was most successful in finding improved solutions to the sector class problems and also found success with random and cluster type problems. Both the savings algorithm and ACO found the urban-rural to be the most difficult problem followed by the coastal class of problems.

Future research should focus on three areas. First, the results for the savings algorithm varied greatly from previously reported results [2][3] and its possible that a parallel algorithm was used in previous research. Further algorithm testing should explore the differences in variations of the savings algorithm. Second, continued study on a larger set of logistics-oriented problems seems worthwhile and will more accurately simulate true market conditions where several hundred stops may be required. Finally, additional metaheuristic algorithms should be tested against logistic-oriented problems to see if certain algorithms have an advantage over the methods used in this study for spatially different demands.

REFERENCES

[1] Agarwal, Y.K. A set partitioning approach to the vehicle routing problem. Ph.D. Dissertation, Case Western University, 1985.

[2] Ballou, R.H. & Agarwal, Y.K. A performance comparison of several popular algorithms for vehicle routing and scheduling. Journal of Business Logistics, 1989, 9 (1), 51-65.

[3] Ballou, R. H. A continued comparison of several popular algorithms for vehicle routing and scheduling. *Journal of Business Logistics,* 1990, 11 (1), 111-126.

[4] Bauer, A., Bullnheimer, B., Hartl, R.F. & Strauss, C. An ant colony optimization approach for the single machine tool tardiness problem. *Proceedings of the 1999 Congress on Evolutionary Computation*, 1999, 1445-1450.

[5] Bell, J.E. & McMullen, P.R. An ant colony optimization approach to the vehicle routing problem. *Advanced Engineering Informatics*, 2004, 18(1), 41-48.
[6] Bland, J.A. Space planning by ant colony optimization. *International Journal of Computer Applications in Technology,* 1999, 12 (6), 320-328.
[7] Bullnheimer, B., Hartl, R.F. & Strauss, C. Applying the ant system to the vehicle routing problem. Voss, S., Martello, S., Osman, I.H. & Roucairol, C. (eds.). *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Boston: Kluwer, 1998.
[8] Bullnheimer, B., Hartl, R.F. & Strauss, C. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research,* 1999, 89, 319-328.
[9] Christofides, N. & Eilon, S. An algorithm for the Vehicle Dispatching Problem. *Operational Research Quarterly,* 1969, 20 (3), 309-318.
[10] Christofides, N., Mingozzi, A. & Toth P. The vehicle routing problem. Christofides, N., Mingozzi, A., Toth, P. & Sandi, C. (eds). *Combinatorial Optimization*, Chichester: Wiley, 1979.
[11] Clark, G. & Wright, J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research,* 1964*,* 12, 568-581.
[12] Dorigo, M. *Optimization, Learning and Natural Algorithms* (In Italian) Unpublished Doctoral Dissertation, Politecnico di Milano, Dipartimento di Elettronica, Italy, 1992.
[13] Dorigo, M. & Gambardella, L.M. Ant colonies for the traveling salesman problem. *BioSystem,* 1997*,* 43 (1), 73-81.
[14] Dorigo, M., Di Caro, G. & Gambardella, L.M. Ant algorithms for discrete optimization. *Artificial Life*, 1999, 5, 137-172.
[15] Gendreau, M., Hertz, A. & Laporte, G. A tabu search heuristic for the vehicle routing problem. *Management Science,* 1994*,* 40, 1276-1290.
[16] Lenstra, J. & Rinnoy Kan,.A. Complexity of vehicle routing and scheduling problems. Networks: 1981, 11: 221-227.
[17] McMullen, P.R. An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering, 2001,* 15, 309-317.
[18] Osman, L.H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 1993, 41, 421-451.