2011 International Conference on Environment Science and Engineering
IPCBEE vol.8 (2011) © (2011) IACSIT Press, Singapore

Review Of Ant Colony Optimization Algorithms OnVehicle Routing Problems And Introduction ToEstimation-Based ACO

NajmeZehraNaqvi, HarmeenKaurMatheru
Deptt.Of Computer Science Engineering, Deptt. Of Computer Science Engineering
IGIT,GGSIPUIGIT,GGSIPU
Delhi,IndiaDelhi,India
naqvizehra@rediffmail.com, harmeen.matheru@gmail.com

KomalChadha
DepttOf Computer Science Engineering
IGIT,GGSIPU
Delhi,India
Komal.chadha88@gmail.com

***Abstract*-Ant colony optimization (ACO) is a meta-heuristic approach to tackle hard combinatorial optimization problems. The basic component of ACO is a solution construction mechanism, which simulates the decision-making processes of ant colonies as they forage for food and find the most efficient routes from their nests to food sources.This paper is a review report on ant colony optimization with its algorithms in chronological order and Vehicle routing problem (one of the application of ACO).Following this, there is a brief introduction of Estimation-based ACO.**

Keywords-Ant Colony Optimization, Metaheuristics, ACO Algorithms, Vehicle Routing Problems, Estimation-based ACO

I. INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems[1]. A Meta heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems[9].

The first algorithm which can be classified within this framework was presented in 1991 by Marco Dorigo with his PHD thesis "Optimization, learning, and Natural Algorithms", modeling the way real ants solve problems using pheromones, and, since then, many diverse variants of the basic principle have been reported in the literature. Real ants are capable of finding the shortest path from a food source to their nest. While walking ants deposit pheromone on the ground and follow pheromone previously deposited by other ants, the essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions[9]. In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is reminiscent of the one adopted by real ants.

To find a shortest path, a moving ants lay some pheromone on the ground, so an ant encountering a previously trail can detect it and decide with high probability to follow it. As a result, the collective behavior that emerges is a form of a positive feedback loop where the probability with which an ant choose a path increases with the number of ants that previously chose the same path[10].

Ant colony optimization is an iterative distributed algorithm. At each iteration, a set of artificial ants (cooperating agents) are considered. At each step of the solution construction, an ant selects the following vertex to be visited according to a stochastic mechanism that is biased by the pheromone: when in vertex i, the following vertex is selected stochastically among the previously unvisited ones. In particular, if j has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge (i, j). At the end of an iteration, on the basis of the quality of the solutions constructed by the ants, the pheromone values are modified in order to bias ants in future iterations to construct solutions similar to the best ones previously constructed[6][11].The ACO system contains two rules:

1. Local pheromone update rule, which applied whilst constructing solutions.
2. Global pheromone updating rule, which applied after all ants construct a solution

Furthermore, an ACO algorithm includes two more mechanisms: trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective[2][3][6].Although each ant of the colony is complex enough to find a feasible solution to the problem under consideration, good quality solutions can only emerge as the result of the collective interaction among the ants. Each ant makes use only of private information and of information local to the node it is visiting.

II. ACO ALGORITHMS

Thereis a non-exhaustive list of successful ant colony optimization algorithms in chronological order as given in Table I[1].

TABLE I. ACO ALGORITHMS

SNo.	Year	Algorithm	Authors
1.	1991	Ant System	Dorigo.et.al
2.	1992	Elitist A.S.	Dorigo.et.al
3.	1995	Ant-Q	Cambardella&Dorigo
4.	1996	Ant Colony System	Cambardella&Dorigo
5.	1996	Max-Min A.S.	Stutzle&Hoos
6.	1997	Rank Based A.S.	Bullnheimer et al
7.	1999	ANTS	Maniezzo
8.	2000	BWAS	Cordon et al
9.	2001	Hyper-Cube A.S.	Blum et al

A.Ant System

Ant system (AS) was the first ACO algorithm to be proposed in the literature (Dorigo et al. 1991, Dorigo 1992, Dorigo et al. 1996)[4]. Its main characteristic is that the pheromone values are updated by *all* the ants that have completed the tour. Solution components $\boldsymbol{c_{ij}}$are the edges of the graph, and the pheromone update for $\boldsymbol{\tau_{ij}}$, that is, for the pheromone associated to the edge joining nodes i and j , is performed as follows:

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} (1)$$

where $\rho \in (0,1]$ is the evaporation rate, mis the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone laid on edge (i,j) by the *k*-th ant:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{L_k} if\ ant\ k\ used\ edge(i,j) in\ its\ tour, \\ 0 \qquad otherwise \end{cases} (2)$$

where L_k is the tour length of the *k*-th ant.

When constructing the solutions, the ants in AS traverse the construction graph and make a probabilistic decision at each vertex. The transition probability $p(c_{ij}|s_k^p)$ of the *k*-th ant moving from node i to node j is given by:

$$p(c_{ij}|s_k^p) = \begin{cases} \tau_{ij}^{\alpha}.\frac{\eta_{ij}^{\beta}}{\sum_{c_{il}\epsilon N(s_k^p)} \tau_{il}^{\alpha}.\eta_{il}^{\beta}} \ if\ j\epsilon N(s_k^p), \\ 0 \qquad otherwise \end{cases} \qquad (3)$$

where $N(s_k^p)$ is the set of components that do not belong yet to the partial solutions_k^p of ant *k*, and α and β are parameters that control the relative importance of the pheromone versus the heuristic information $\eta_{ij} = 1/d_{ij}$, whered_{ij} is the length of componentc_{ij} (i.e., of edge (i,j)).

A. Ant colony system

The first major improvement over the original ant system to be proposed was ant colony system (ACS), introduced by Dorigo and Gambardella (1997)[4]. The first important difference between ACS and AS is the form of the decision rule used by the ants during the construction process. Ants in ACS use the so-called*pseudorandom proportional* rule: the probability for an ant to move from node i to node j depends on a random variable q uniformly distributed over [0,1], and a parameter q_0; if q≤ q_0, then, among the feasible components, the component that maximizes the product$\tau_{il}.\eta_{il}^{\beta}$ is chosen; otherwise, the same equation as in AS is used.

This rather greedy rule, which favors exploitation of the pheromone information, is counterbalanced by the introduction of a diversifying component: the *local pheromone update*. The local pheromone update is performed by all ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1-\varphi).\tau_{ij} + \varphi.\tau_0, (4)$$

where $\varphi \in (0,1]$ is the pheromone decay coefficient, andτ_0 is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during one iteration. In fact, decreasing the pheromone concentration on the edges as they are traversed during one iteration encourages subsequent ants to choose other edges and hence to produce different solutions. This makes less likely that several ants produce identical solutions during one iteration. Additionally, because of the local pheromone update in ACS, the minimum values of the pheromone are limited.

As in AS, also in ACS at the end of the construction process a pheromone updatecalled *offline* pheromone update, is performed.

ACS offline pheromone update is performed only by the best ant, that is, only edges that were visited by the best ant are updated, according to the equation:

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \rho.\Delta\tau_{ij}^{best} (5)$$

where $\Delta\tau_{ij}^{best} = 1/L_{best}$ if the best ant used edge (i,j) in its tour$\Delta\tau_{ij}^{best} = 0$, otherwise ($L_{best}$ can be set to either the length of the best tour found in the current iteration -- *iteration-best*, L_{ib} -- or the best solution found since the start of the algorithm -- *best-so-far*,L_{bs}).It should be noted that most of the innovations introduced by ACS were introduced first in Ant-Q, a preliminary version of ACS by the same authors.

B. MAX-MIN ant system

MAX-MIN ant system (MMAS) is another improvement, proposed by Stützle and Hoos (2000), over the original ant system idea. MMAS differs from AS in that (i) only the best ant adds pheromone trails, and (ii) the minimum and maximum values of the pheromone are explicitly limited (in AS and ACS these values are limited implicitly, that is, the value of the limits is a result of the algorithm working rather than a value set explicitly by the algorithm designer)[4].The pheromone update equation takes the following form:

$$\tau_{ij} \leftarrow (1-\rho).\tau_{ij} + \Delta\tau_{ij}^{best} \qquad (6)$$

where$\Delta\tau_{ij}^{best} = 1/L_{best}$ if the best ant used edge (i,j) in its tour,$\Delta\tau_{ij}^{best} = 0$ otherwise, whereL_{best} is the length of the tour of the best ant. As in ACS, L_{best} may be set (subject to the algorithm designer decision) either toL_{ib} or toL_{bs} , or to a combination of both.The pheromone values are constrained between τ_{min} and $\tau_{max}$by verifying, after they have been updated by the ants, that all pheromone values are within the imposed limits: τ_{ij} is set toτ_{max} if $\tau_{ij} > \tau_{max}$ and toτ_{min} if$\tau_{ij} < \tau_{min}$. It is important to note that the pheromone update equation of MMAS is applied, as it is the case for AS, to all the edges while in ACS it is applied only to the edges visited by the best ants.

The minimum valueτ_{min} is most often experimentally chosen (however, some theory about how to define its value analytically has been developed in (Stützle&Hoos 2000)). The maximum value τ_{max} may be calculated analytically provided that the optimum ant tour length is known.

III. VEHICLE ROUTING PROBLEMS

The Vehicle Routing Problem concerns the transport of items between depots and customers by means of a fleet of vehicles. In general, solving a VRP means to find the best route to service all customers using a fleet of vehicles.The solution must ensure that all customers are served, respecting the operational constraints, such as vehicle capacity and the driver's maximum working time, and minimizing the total transportation cost.A VRP can be formulated as a mathematical programming problem, defined by an objectivefunction, and a set of constraints. Exploiting the characteristics of the mathematical formulation of the problem, we want to design an algorithm able to efficiently find a solution. The problem can be formulated as a graph theoretic problem, where G = (V,A) is a complete graph, V the vertex set (customers, and the depot, usually labeled with 0) and A is the arc set (the paths connecting all customers and the depot). A non-negative demand q_i is associated with each vertex, and a cost c_{ij} is associated with each edge in A.

A. Basic problems of the vehicle routing class

Combining the various elements of the problem, we can define a whole family of different VRPs[13].We briefly introduce the Capacitated VehicleRouting Problem (CVRP), the VRP with Time Windows (VRPTW) and its Time Dependentvariant (TDVRPTW), the VRP with Pickup and Delivery (VRPPD), and the Dynamic VRP(DVRP).

B. The Capacitated VRP

The Capacitated Vehicle Routing Problem (CVRP) is the basic version of the VRP. The name derives from the constraint of having vehicles with limited capacity[14]. In the classic version of the CVRP, customer demands are deterministic andknown in advance. Deliveries cannot be split, that is, an order cannot be served using two or more vehicles. The vehicle fleet is homogeneous and there is only one depot. The objective is to minimize total travel cost, usually expressed as the travelled distance required to serve all customers.If the cost matrix associated with the graph, representing the distance or travel time, is asymmetric, then the problem is called the asymmetric CVRP.

C. VRP with Time Windows

In a Vehicle Routing Problem with Time Windows (VRPTW) the capacity constraint still holds and each customer i is associated with a time interval $[a_i,b_i]$, called the time window, and with a time duration,s_i, the service time. Time windows can be set to any width, from days to minutes, buttheir width is often empirically bound to the width of the planning horizon. The presence of time windows imposes a series of precedence on visits, which make the problem asymmetric, even if the distance and time matrices were originally symmetric.VRPTW is also NP-hard and even to find a feasible solution to VRPTW is an NP-hardproblem[15]. The additional constraints in VRPTW call for more articulate variants of the basic methods used to obtain an exact solution for CVRP and therefore the performancestend to worsen.

D. VRP with Pick-up and Delivery

In VRP with pick-up and delivery (VRPPD) a vehicle fleet must satisfy a set of transportation requests. The transport items are not originally concentrated in the depots, but they are distributed over the nodes of the road network. A transportation request consists in transferring the demand from the pick-up point to the delivery point. These problems always include time windows for pick-up and/or delivery and also constraints that express the userinconvenience of waiting too long at the pick-up point and impose a limit on riding time. Whenthe demand is a transport of goods, sometimes the problem can be simplified, according to thecharacteristics of the transport process. All deliveries can be performed before the pick-ups, thus reducing the impact of capacity constraints[16].

E. Time Dependent VRP

An extension of the VRPTW in urban environments is the Time Dependent VRPTW, where the arc costs on the graph depend on time.The time taken to travel from a location to another one depends on the traffic load, which varies with the time of the day. Particular care must be taken indefining the time dependency of the cost function. If the horizon of interest is quantized into small intervals, and the travel times vary in discrete jumps from an interval to the next, then this approach, although being quite used, does produce solutions which may go against common-sense. This happens when the FIFO property is violated, thatis, a vehicle departing later may arrive earlier than an earlier departing vehicle, even following the same route. Therefore formulations where travel time and cost functions vary continuously are to be preferred[17].

F. Dynamic VRP

When the service requests are not completely known before the start of service, but they arrive during the distribution process. This variant is called Dynamic Vehicle Routing Problem (DVRP). Since new orders arrive dynamically, the routes have to be replanned at run time in order to include them. Every driver has, at each time step, a partial knowledge about the remainder of his/her tour.Among possible applications of DVRP we find feeder systems, which typically are localdial-a-ride systems aimed at feeding another, wider area, transportation system at a particular transfer location[18]. Another application is to courier service problems (e.g. Federal Express), where parcels are collected at customer locations and brought back to a central depot for further processing and shipping.

IV. ESTIMATION-BASED ACO

Here, we study the application of ACO algorithms to the PROBABILISTIC TRAVELING SALESMAN PROBLEM (PTSP) (Jaillet 1985), which is also known as the traveling salesman problem with stochastic customers (Gendreau et al. 1996)[19]. It is a stochastic extension of the classical traveling salesman problem (TSP). In the PTSP, it is unknown in advance whether a node requires being visited, but its probability of requiring a visit is given. The most widely used approach to tackle the PTSP is to construct an a priori solution before knowing which nodes require being visited. An a priori solution is a permutation of all the nodes of the given instance. Once the set of nodes that require being visited is known, then a posteriori solution is derived by visiting the nodes that require being visited in the order prescribed by the a priori solution and by skipping the nodes that do not require being visited. The objective of the PTSP is to find an a priori solution, such that the expected cost of its associated a posteriori solution is minimized.

The PTSP is an *NP*-hard problem. The stochastic nature and the complexity of the problem limit the applicability of exact algorithms. Recent approaches to the PTSP mainly involve the application of stochastic local search (SLS) methods (Hoos and Stützle 2005), among which ACO algorithms appear to be currently the best-performing. SLS methods for the PTSP can be grouped into two main classes: *analytical computation* and *empirical estimation*. In analytical computation algorithms, the exact cost of an a priori solution is computed using a closed-form expression derived by Jaillet (1985). A prominent subclass of analytical computation algorithms is analytical approximation, where the cost of the a priori solution is approximated using a truncated version of the closed-form expression. In empirical estimation algorithms, the cost of the a priori solution is estimated by Monte Carlo simulation.

Much of the early ACO algorithms for the PTSP are based on analytical computation. Bianchi et al. (2002a, 2002b) adopted the closed-form expression in an ant colony system (ACS) and compared it with a version of ACS for the TSP. The preliminary results showed that the PTSP-specific approach is more effective than its TSP counterpart when the instance probability values are less than 0.5. Branke and Guntsch (2004) explored the idea to employ an ad-hoc approximation to replace the exact PTSP objective function, and showed that the computation time can be significantly reduced without major loss in solution quality. Bianchi (2006)[23] and Bianchi and Gambardella (2007)[22] proposed pACS+1-shift, which integrates the PTSP-specific ACS with 1-shift (Bertsimas and Howell 1993; Bianchi et al. 2005), a local search tailored for the PTSP. The experimental results showed that pACS+1-shift significantly outperforms all other algorithms proposed so far in the literature, and it is up to now considered as the best-performing SLS method for the PTSP.

In our recent research on the PTSP, we first developed 2.5-opt-EEs (Birattari et al. 2008)[20], a new local search algorithm that uses an estimation-based approach to speed up the cost difference computation between neighbor solutions. 2.5-opt-EEs reaches significantly better solutions for a wide range of instances of size from 100 to 1000 nodes and it is by two to three orders of magnitude faster than 1-shift. Only on instances with low probability values, 2.5-opt-EEs showed a slightly worse performance than 1-shift. To address this issue, we integrated two variance reduction techniques, namely adaptive sample size and importance sampling into 2.5-opt-EEs, obtaining in this way the new algorithm 2.5-opt-EEais (Balaprakash et al. 2009)[19][21], which fully outperforms 1-shift. To test whether the observed behavior extends to SLS methods, we performed some preliminary experiments with a simple iterated local search algorithm that uses 2.5-opt-EEais or an improved variant of 1-shift. The results showed that the iterated local search algorithm with 2.5-opt-EEais is very effective with respect to both solution quality and computation time (Balaprakash et al. 2009)[19]. These results indicate that there is also a significant potential to improve over pACS+1-shift. The aforementioned factors motivated us to develop a new algorithm that adopts the estimation-based approach in the ACO framework, with the goal of effectively solving the PTSP.

A. The pACS+1-shift algorithm

pACS+1-shift (Bianchi 2006; Bianchi and Gambardella 2007)[22][23] is currently the best performing ant colony optimization algorithm for the PTSP. It is a standard ACS algorithm (Dorigo and Gambardella 1997) in which, at each iteration, m ants construct solutions in the following way: With a probability q_0, ant k at node i chooses to move to the node j that maximizes the product $\tau_{ij}\eta_{ij}^{\beta}$; with probability $1-q_0$, the next node j is chosen with probability

$$p_{ij}^{k} = \tau_{ij}\eta_{ij}^{\beta} / \sum_{l\in N_i^k} \tau_{il}\eta_{il}^{\beta} \quad (7)$$

(the random proportional rule); τ_{ij} and $\eta_{ij} = 1/c_{ij}$ are

the pheromone value and the heuristic value associated with edge <*i,j*>, respectively; β is a parameter that determines the relative influence of the heuristic information; N_i^k is the set of nodes for which it is feasible to move from node i. When an ant moves from node i to node j , the pheromone value associated with edge <*i, j* >is updated to

$$\tau_{ij} = (1-\varphi).\tau_{ij} + \varphi.\tau_0 \quad (8)$$

where $\varphi \in (0,1]$ is a parameter, and τ_0 is the initial value of the pheromone. At the end of each iteration, the pheromone value associated with each edge <*i,j*>of the best-so-far solution is updated to

$$\tau_{ij} = (1-\rho).\tau_{ij} + \rho.\Delta\tau_{ij}^{best} \quad (9)$$

where $\rho \in (0,1]$ is a parameter and $\Delta\tau_{ij}^{best} = 1/C^{best}$. The value of C^{best} is set to the cost of the best-so-far solution. 1-shift local search, a PTSP-specific iterative improvement algorithm, is applied to all solutions constructed by the ants prior to the pheromone update. The algorithm proceeds in two phases: The first phase consists in exploring a swap-neighborhood, where the set of neighbors of a given solution contains all the solutions that can be obtained by swapping two consecutive nodes. The second phase explores the node-insertion neighborhood in a fixed lexicographic order. The cost difference of neighboring solutions is obtained by delta evaluation, a technique that considers only the cost contribution of solution components that are not common between the two solutions. This is done using computationally expensive closed-form expressions, which are based on complex mathematical derivations (Bianchi et al. 2005; Bianchi 2006; Bianchi and Campbell 2007)[22].

B. Effectiveness of 2.5-opt-EEais in pACS

2.5-opt-EEais (Balaprakash et al. 2009)[19][21] is the state-of-the-art iterative improvement algorithm for the PTSP. 2.5-opt-EEais differs from 1-shift in the following three elements: It adopts an empirical estimation technique in the delta evaluation; it uses the 2.5-exchange neighborhood relation that combines the 2-exchange and node-insertion neighborhoods (Bentley 1992); and it exploits typical TSP neighborhood reduction techniques such as fixed-radius search, candidate lists, and don't look bits (Martin et al. 1991; Bentley 1992; Johnson and McGeoch 1997).

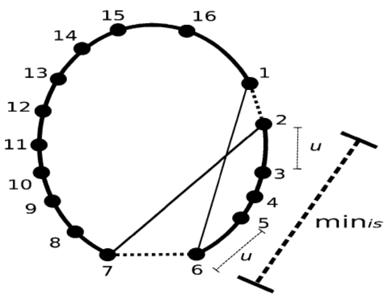

Fig 1.In this example, the two edges <1,2>and <6,7>are deleted and replaced with <1,6>and <2,7>by a 2-exchange move. Assume that $min_{is}$and u are set to 50 and 40, respectively. Since the number of nodes in the segment [2, . . . , 6] is less than 50% of 16, which is eight, importance sampling is used to bias 40% of 5, that is, two nodes on *each end* of the segment [2, . . . , 6]: on the end that starts with node 2, the biased nodes are 2 and 3; on the other end that starts with node 6, the biased nodes are 5 and 6

The effectiveness of the algorithm is further enhanced by the usage of variance reduction techniques such as the method of common random numbers, adaptive sample size, and importance sampling. In particular, importance sampling is essential for the algorithm to effectively tackle instances with probability values up to 0.2. Moreover, the adoption of this procedure is useful for instances with high probability values although it may result in slightly higher computation time when compared to an appropriate fixed sample size of 100 (Balaprakash et al. 2009)[21]. The importance sampling procedure is implemented as follows: In a 2-exchange move, whenever the number of nodes in the shorter segment (a 2-exchange move always leads to two tour segments) is less than minis% of the instance size, u% nodes on *each end* of the shorter segment are biased with probability p'See Fig. for an example. Whenever a node i is biased with ty p' ,the delta evaluation procedure ignores realizations sampled with the original probability $p_i$and considers instead realizations in which the probability of node i requiring being visited is p'. The cost difference estimate obtained in this way is a biased one, which is then corrected to an unbiased one using the likelihood ratio. For the node-insertion move, only the insertion node is biased with a value p''. For a more detailed explanation of 2.5-opt-EEais, we refer the reader to Balaprakash et al. (2009).We denote pACS+2.5-opt-EEais to be the algorithm obtained by combining pACS with 2.5-opt-EEais.

V. CONCLUSION AND FUTURE WORK

This paper presents the approach of ACO algorithm implementation and all its versions. This ACO approach is used on one of its applications VRP, and its variants (CVRP, VRPTW, TD-VRPTW, VRPPD, DVRP). We used the current best performing ACO algorithm pACS+1-shift as a starting point. We showed that the adoption of the state of-the-art iterative improvement algorithm 2.5-opt-EEais allows pACS to obtain a significant improvement in the solution cost. To develop a complete estimation-based ACS, we adopted an estimation-based approach to evaluate the solution costs. The future work is focused to design estimation-based SLS methods to solve stochastic vehicle routing problems.

REFERENCES

[1] "Ant Colony Optimization"IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008

[2] H.R. Lourenco, O. Martin, and T. Stutzle, "Iterated Local Search", in Handbook of Metaheuristics, ser. International Series in Operations Research & Management Science, F. Glover and G.Kochenberger, Eds., Kluwer Academic Publishers, vol. 57, pp. 321-353, 2002.

[3] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithm for discrete optimization", Artificial Life, vol. 5, no. 2, pp. 137-172, 1999.

[4] Marco Dorigo,ThomasStuttzle" Ant Colony Optimization" Massachusetts Institute of Technology,2004

[5] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O Rossi-Doria, and T. Schiavinotto. Metaheuristics for the vehicle routing problem with stochasticdemands. Technical Report TR-12-04, IDSIA, Galleria 2, Manno, 6928, Switzerland, 2004.

[6] M. Dorigo, M. Birattari, and T. Stitzle, "Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique, IEEE computational intelligence magazine, November, 2006.

[7] P. Jungie, W. Dingwei, "An Ant Colony Optimization Algorithm for Multiple Treavelling Salesman Problem", ICICIC apos; 06. first international conference on Volume 1, Issue, 30-01 Aug. 2006, page(s) : 210-213.

[8] M. Caramia, G.F. Italiano, G. Oriolo, A. Pacifici, and A. Perugia. Routing a fleet of vehicles for dynamic combined pick-up and deliveries services. Universit´a di Roma "Tor Vergata", 2002.

[9] M. Dorigo, luca M. G., " Ant Colony system: A Cooperative learning approach to the Travelling Salesman Problem", IEEE transaction on evolutionary computation, Vol. 1, No. 1, 1997.

[10] M. Dorigo and G. Di Caro, "The Ant Colony Optimization metaheuristic", in New Ideas in Optimization, D. Corne et al., Eds., McGraw Hill, London, UK, pp. 11-32, 1999.

[11] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant algorithm for discrete optimization", Artificial Life, vol. 5, no. 2, pp. 137-172, 1999.

[12] H.R. Lourenco, O. Martin, and T. Stutzle, "Iterated Local Search", in Handbook of Metaheuristics, ser. International Series in Operations Research & Management Science, F. Glover and G.Kochenberger, Eds., Kluwer Academic Publishers, vol. 57, pp. 321-353, 2002.

[13] P. Toth and D. Vigo. The Vehicle Routing Problem, chapter An Overview of Vehicle Routing Problems, pages 1–26. SIAM, Society for Industrial and Applied Mathematics,Philadelphia, USA, 2000.

[14] P. Toth and D. Vigo. Fleet Management and Logistics, chapter Exact solution of the vehicle routing problem, pages 1–31. Kluwer Academic Publishers: Boston/Dordrecht/London, 1998.

[15] M.W.P. Savelsbergh. Local search in routing problems with time windows. Annals of Operations Research, 4:285–305, 1985.

[16] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon, and F. Soumis. The Vehicle Routing Problem, chapter VRP with Pickup and Delivery, pages 225–242. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, USA, 2000.
[17] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research, 144(2):379–396, 2003.
[18] P. Kilby, P. Prosser, and P. Shaw. Dynamic VRPs: a study of scenarios. Technical Report APES-06-1998, University of Strathclyde, September 1998.
[19] PrasannaBalaprakash · Mauro Birattari ·Thomas Stützle · Zhi Yuan · Marco Dorigo" Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem",June 2009
[20] Balaprakash, P., Birattari, M., Stützle, T., Yuan, Z., &Dorigo, M. (2008). Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem. IRIDIA Supplementarypage.
[21] Balaprakash, P., Birattari, M., Stützle, T., &Dorigo, M. (2009). Adaptive sample size and importance sampling in estimation-based local search for the probabilistic traveling salesman problem. European Journal of Operational Research, 199(1), 98–110.
[22] Bianchi, L., & Campbell, A. (2007). Extension of the 2-p-opt and 1-shift algorithms to the heterogeneous probabilistic traveling salesman problem. European Journal of Operational Research, 176(1), 131–144.
[23] Bianchi, L. (2006). Ant colony optimization and local search for the probabilistic traveling salesman problem: a case study in stochastic combinatorial optimization. Ph.D. thesis, UniversitéLibre de Bruxelles,Brussels, Belgium.