A Hybrid Ant Colony Optimisation Technique for Dynamic Vehicle Routing

Darren M. Chitty and Marcel L. Hernandez

The Advanced Processing Centre, QinetiQ Ltd.,
St Andrews Road, Malvern,
Worcestershire, WR14 3PS, UK
{chitty, marcel}@signal.qinetiq.com

Abstract. This paper is concerned with a dynamic vehicle routing problem. The problem is dynamic in the sense that the time it will take to traverse each edge is uncertain. The problem is expressed as a bi-criterion optimisation with the mutually exclusive aims of minimising both the total mean transit time and the total variance in transit time. In this paper we introduce a hybrid dynamic programming - ant colony optimisation technique to solve this problem. The hybrid technique uses the principles of dynamic programming to first solve simple problems using ACO (routing from each adjacent node to the end node), and then builds on this to eventually provide solutions (i.e. Pareto fronts) for routing between each node in the network and the destination node. However, the hybrid technique updates the pheromone concentrations only along the first edge visited by each ant. As a result it is shown to provide the overall solution in quicker time than an established bi-criterion ACO technique, that is concerned only with routing between the start and destination nodes. Moreover, we show that the new technique both determines more routes on the Pareto front, and results in a 20% increase in solution quality for both the total mean transit time and total variance in transit time criteria. However the main advantage of the technique is that it provides solutions in routing between each node to the destination node. Hence it allows "instantaneous" re-routing subject to dynamic changes within the road network. [1]

1 Introduction

A requirement in the routing of a single vehicle through a road network from a starting depot to a destination depot, is the ability to manoeuver quickly to take into account events such as blocked roads or heavy traffic. Subsequently, the problem becomes dynamic because the road conditions are not known with certainty and are continuously changing. As a result, each road is characterised by two indices. The first of these is the mean transit time, averaged over different driving scenarios. The second is the variance in transit time on each road, which gives an indication of how the transit time will fluctuate about this mean value as the scenario changes.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 48–59, 2004.

In an ideal world, one would wish to find routes that have both low mean transit time and low variance in transit time. However, typically these priorities have conflicting objectives. Routes that have the shortest overall travel time may not have the smallest variance in travel time and vice-versa. In such circumstances we must then trade-off between these two conflicting aims.

This is the basis of bi-criterion optimisation (see [1]). Instead of attempting to find a solution that satisfies the minimisation of each objective, we seek out the set of non-dominated solutions that form the Pareto front in the two-dimensional objective function space. Evolutionary techniques, which simultaneously create and evaluate a set of possible solutions, are a natural approach to solving problems of this type (again, see [1]). Alternative techniques, such as linear programming, have also been used for multi-objective optimisation in which one objective is minimised with (worst acceptable) performance bounds placed on each of the other objectives [2].

In this paper, we build on a technique known as Ant Colony Optimisation [3], [4] to solve this bi-criterion routing problem. For a variety of reasons, Ant Colony Optimisation is a natural approach. Firstly, its foundations lie in the way in which real ant colonies find shortest path routes between different parts of their natural habitat [5]. As a result, the technique has been successfully proved to be particularly effective in solving networking problems.

Indeed, it has been successfully applied to the Travelling Salesman Problem (TSP) [3], [4]; the Graph Colouring Problem [6]; and the Vehicle Routing Problem [7], [8]. The Vehicle Routing Problem considered in [7] and [8] is different from the one analysed in this report in that these papers are concerned with the routing of a fleet of vehicles to satisfy a number of customer requests. The vehicles begin and end at a central depot, and once the customers are assigned to vehicles the Vehicle Routing Problem is reduced to several Travelling Salesman Problems.

In each of these cases, the model is deterministic, but Ant Colony Optimisation has also been applied to several dynamic problems. It has been used for routing in communication networks [9], [10] in which there is uncertain demand on each node, with requests forming a dynamic and uncertain sequence. Moreover, ant techniques have also been applied successfully to a dynamic Travelling Salesman Problem in which, at certain time instances, parts of the network are 'lost' and re-routing is necessary [11]. The ability of the technique to cope with problems of this type makes it a natural approach to both vehicle routing, and more generally, to solving problems that are subject to dynamic and uncertain change.

2 The Bi-criterion Optimisation Problem

Consider a road network repesented by $G = (N, E)$, where $N = (N_1, ...N_n)$ is the set of n nodes (i.e. junctions) and E is the set of (directed) edges (i.e. roads, where a direction of travel may be specified). The aim is to route vehicles so that they will reach their destination in the quickest time possible. However, the problem is subject to uncertainty. Traffic congestion may cause delays, other forms of disruption such as road works and/or driving accidents, may also affect transit times. As a result, we are unable to simply characterise each edge, E_{ij}, in terms of the time it will take to traverse.

However, we can instead characterise E_{ij} in terms of two indices; these being the average time: M_{ij}, and the variance in time: V_{ij}, it will take to traverse each edge. M_{ij}, is averaged over the different driving scenarios. The variance, V_{ij}, gives an indication of how the travel time will fluctuate about the mean value as the scenario changes.

We will specify each route as $R = (a_{ij})$ where:

$$a_{ij} = \begin{cases} 1 \text{ if node } j \text{ is visited after node } i \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

for $i, j = 1, \ldots, n$.

It is assumed that the time taken to traverse edge E_{ij} is independent of the time taken to traverse each of the other edges. This assumption is not entirely true, and indeed delays on certain edges may be expected to have a knock-on effect on other edges within the network. However, provided the network is not densely congested, these effects are likely to be small and make this assumption a valid approximation. It then follows that the average total transit time, $T_m(R)$, and the variance in total transit time, $T_v(R)$, are given by:

$$T_m(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}M_{ij} \qquad T_v(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}V_{ij} \tag{2}$$

Clearly, if mean transit time is proportional to variance in transit time, i.e.

$$M_{ij} \propto V_{ij}^{k} \qquad \text{for some } k > 0 \tag{3}$$

then by determining a route, R, that minimises mean time (equation (2)) the variance in time (equation (3)) is also minimised. In this case we have effectively just a single objective function, and the optimum route can easily be found using a Dynamic Programming technique such as Dijkstra's algorithm [12].

However, in general we would not expect a relationship as simple as (4) to hold. Indeed, with small average transit times may typically represent urban routes, that can usually be traversed quickly, but can be easily disrupted by traffic congestion. Conversely, edges with large mean transit times may represent long motorway segments, which are designed to be less prone to traffic disruption.

With this in mind, a more suitable relationship between M_{ij} and V_{ij} is given by:

$$M_{ij} \propto \frac{1}{V_{ij}^{k}} \qquad \text{for some } k > 0 \tag{4}$$

This is the relationship that we will assume in later examples. Clearly now if we minimise $T_m(R)$ we will maximise $T_v(R)$ and, indeed the converse is also true. We are therefore faced with a bi-criterion optimisation problem (again, see [1]).

The general structure of the solution space is shown in figure 1. Instead of attempting to find a solution that simultaneously minimises each objective, which is clearly no longer possible, we seek the set of non-dominated solutions that form the Pareto front in the multi-dimensional objective function space. In figure 1 this Pareto front is represented by the solid thick black edge of the solution space. Any solution not on the Pareto front (i.e. within the red region) is 'dominated' by a solution on the Pareto front which has both lower mean and variance (in transit time) and is clearly better in every respect. Solutions on the Pareto front itself cannot dominate each other, and the solution that should be utilised depends on the scenario and the operational requirements (i.e. we may not accept a solution that has a variance $T_v(R) > V_{max}$, say).

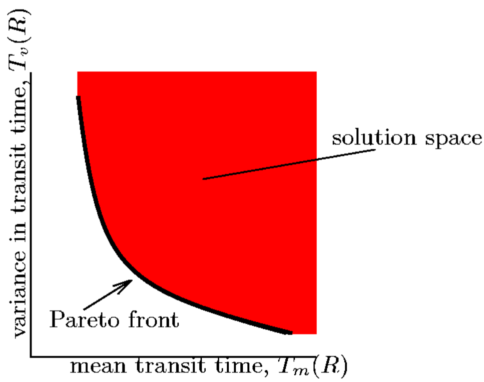

Fig. 1. The general structure of the solution space.

Techniques, such as Ant Colony Optimisation [3], [4], which simultaneously create and evaluate a set of possible solutions are a natural approach to solving problems of this type. Indeed, Ant Colony Optimisation has been used to solve several multi-criterion problems (see [13] for an overview) and, furthermore [13] introduces a general framework for solving bi-criterion optimisation problems that is utilised in this report.

3 Ant Colony Optimisation

Ant Colony Optimisation (ACO) [3], [4] is an evolutionary approach that is inspired by the way in which real ant colonies establish shortest path routes between their nest and feeding sources. Real ants establish such paths by depositing an aromatic essence known as pheromone along these paths [5]. The quantity of pheromone is proportional to the length of the path, or the quality of the feeding habitat. Ants are attracted to the pheromone and follow it. Hence, the pheromone concentrations along better paths will be further enhanced which will attract more ants. Eventually, the pheromone concentrations along better paths will become so great that all ants will uses these routes.

Like all evolutionary techniques, Ant Colony Optimisation provides a means of exploring promising areas of the solution space in order to find optimal/near optimal solutions. It does this by creating a trade-off between the conflicting aims of exploitation and exploration, where

- exploration is defined to be the search through the space of possible solutions in order to find the optimal/near optimal solution(s);
- exploitation is the fusion of information regarding the quality of solutions found so far, in order to focus on promising areas of the search space.

In addition to the natural application of ant algorithms to networking problems, they have been successfully applied to a host of other combinatorial optimisation problems (see [10] for an overview), including the Quadratic Assignment Problem [14] and the Job Shop Scheduling Problem [15]. Encouraging results have also been obtained when using Ant Colony Optimisation to build up classification rules for data mining [16].

A brief introduction to the technique follows, using an ant algorithm applied to a single objective problem. We will then show how the pheromone matrix can be adapted for a bi-criterion problem.

The basis of Ant Colony Optimisation is the pheromone matrix, $M = \tau_{ij}$. The probability, p_{ij} that $a_{ij} = 1$ (see equation (3-1)) is a function of the amount of pheromone τ_{ij} on 'edge' E_{ij} and information ν_{ij}, relating to the quality of this edge, i.e.

$$p_{ij} = \frac{\tau_{ij}^{\alpha}\nu_{ij}^{\beta}}{\sum_{h \epsilon S} \tau_{ih}^{\alpha}\nu_{ih}^{\beta}} \tag{5}$$

where α, β give the influence of the pheromone and heuristic information respectively; and S is the set of nodes not already visited.

Referring once again to a single objective vehicle routing problem, if the objective is to minimise the mean total transit time, then the heuristic information may be given as follows:

$$\nu_{ij} = 1/M_{ij} \qquad \text{for } i, j = 1, \ldots, n \tag{6}$$

i.e. the preferred edges are those that have the smallest mean transit time. The transition probabilities, p_{ij}, are then used to build candidate solutions, which are referred to as 'ants'. In turn, the quality of each solution determines the way in which the pheromone matrix is updated. 'Paths' that form part of high quality solutions have their pheromone levels reinforced and this enables promising areas of the search space to be identified and explored in subsequent iterations (generations). This is the analogy with real ant systems that reinforce pheromone concentrations along the better paths, leading the subsequent ants towards optimal or near optimal routes.

Pheromone update typically has an evaporation rate, ρ, and a deposit rate Δ_{ij}, i.e.

$$\tau_{ij} \rightarrow (1-\rho)\tau_{ij} + \Delta_{ij} \tag{7}$$

The value of Δ_{ij} is dependent on whether the ant(s) used edge E_{ij}, i.e. whether $a_{ij} = 1$ or $a_{ji} = 1$ and how optimal the overall solution(s) were that used edge E_{ij}. Hence, in the case of attempting to minimise total mean time, if we define:

$$d_{ij}^{r} = \begin{cases} 1 \text{ if edge } E_{ij} \text{ is used by ant } r \text{ (i.e. } a_{ij}^{r}{=}1 \text{ or } a_{ji}^{r}{=}1) \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

then we may take (as in [4]):

$$\Delta_{ij} = \sum_{r=1}^{M} \frac{d_{ij}^{r}}{T_m(r)} \qquad (= \Delta_{ji}) \tag{9}$$

where (to remind the reader), $T_m(r)$ is the total mean transit time of the route, a_{ij}^{r} taken by the ant r and M is the total number of ants in each generation.

Other pheromone update rules have also been proposed. For example, in some cases good results have been obtained by having an additional contribution from the highest quality solutions found so far (so called 'elitist') ants (see [8] and references therein).

The initial pheromone concentrations are usually set at some arbitrarily small value. The stopping criterion is typically either to terminate the ant algorithm if there has been no improvement in the best solution for a fixed number of generations; or we have reached the maximum number of generations permitted.

Clearly, the pheromone evaporation rate, p, provides one means by which the ant algorithm can control the trade-off between exploration and exploitation. If p has a large value (i.e. close to 1) then in each generation of ants, the pheromone matrix is highly dependent on the good solutions from the previous generation, which leads to high degree of search around these good solutions. The smaller the value of p, the greater the contribution of good solutions from all the previous generations and the greater the diversity of search through the solution space.

4 A Standard Bi-criterion Ant Model

We follow the approach of [13] and use two types of pheromone, one relating to the first objective of minimising the total mean transit time ($M_m = (\tau_{ij}^{m})$) and the second matrix relating to the second objective of minimising the variance in the transit time, ($M_v = (\tau_{ij}^{v})$). Each ant then uses the following transition probabilities:

$$p_{ij} = \frac{(\tau_{ij}^{m})^{\lambda}(\tau_{ij}^{\nu})^{(1-\lambda)}}{\sum_{h \epsilon S}(\tau_{ih}^{m})^{\lambda}(\tau_{ih}^{\nu})^{(1-\lambda)}} \tag{10}$$

$\lambda \in [0, 1]$ is the importance of objective one in relation to objective two. If $\lambda = 1$, then the single objective is to minimise the mean transit time, if $\lambda = 0$, the single objective is to minimise the variance in transit time. For value of λ between these two extreme values we must trade off between these two conflicting objectives.

Pheromone update is again as follows:

$$\tau_{ij}^{m} \to (1-\rho)\tau_{ij}^{m} + \sum_{r=1}^{M} \frac{d_{ij}^{r}}{T_m(r)} \qquad \tau_{ij}^{\nu} \to (1-\rho)\tau_{ij}^{\nu} + \sum_{r=1}^{M} \frac{d_{ij}^{r}}{T_v(r)} \tag{11}$$

where d_{ij}^{r} is given by equation (9). Only ants that reach the destination node update the pheromone matrix.

5 The Hybrid Ant Colony Optimisation Technique

5.1 Background

The standard ACO technique of the previous section can give good results for routing vehicles between a start node and a destination node. However, the time taken to produce a solution can be in excess of one minute[2], even for just a 100 node problem instance.

In addition, the technique determines a Pareto front of solutions only for the focal problem of routing from the start node to the destination node. If, for any unforeseen reason, a vehicle has to divert to another node not on the prescribed route (i.e. to avoid a road blocked as a result of an accident), it would no longer have any information that could be used to navigate it to its final destination. A new method is proposed which will lead to a quicker build up of the pheromone on edges that are important and will result in every node in the network having a Pareto front of non-dominated solutions. Hence, instantaneous re-routing can be performed if the target has to deviate from the original plan.

5.2 Features

The key features of this new technique are as follows:

- There are again two types of pheromone ($M_m = (\tau_{ij}^{m})$ and $M_v = (\tau_{ij}^{v})$).
- We use two kinds of ants, one for each criterion.
- Ants optimising the mean transit time criterion use the following transition probabilities:

$$p_{ij} = \frac{\tau_{ij}^{m}}{\sum_{h \epsilon S} \tau_{ih}^{m}} \tag{12}$$

  with a similar expression giving the transition probabilities of ants optimising the variance in transit time criterion.

[2] When using C++ Version 6.0 run on a 600 MHz Athlon processor.

- Under each criterion, M ants travel to the destination node from each and every node in the network.
- Each ant travelling from node i updates a Pareto front, P_i of solutions (routes to the destination node) from that start location.

5.3 Pheromone Update

In each generation, pheromone evaporation occurs at a constant rate ρ. Hence, for the two pheromone matrices:

$$\tau_{ij}^{m} \rightarrow (1-\rho)\tau_{ij}^{m} \qquad \tau_{ij}^{\nu} \rightarrow (1-\rho)\tau_{ij}^{\nu} \tag{13}$$

Pheromone update is performed in the following way.

- Ants only update their own type of pheromone.
- Only solutions on the global Pareto front, P_i (at each node, i) update the pheromone matrices.
- Pheromone update is performed only on the very first path taken by each ant.

Hence, for the rth ($r = 1, \ldots, M$) ant starting from node i and optimising mean transit time, pheromone update is then given by:

$$\tau_{ij}^{m} \rightarrow \tau_{ij}^{m} + \frac{\hat{d}_{ij}^{r}}{T_{m_{im}}(r)} \tag{14}$$

where

$$\hat{d}_{ij}^{r} = \begin{cases} 1 \text{ if: } \begin{cases} \text{edge } E_{ij} \text{ is used by ant } r \\ \text{node } j \text{ is adjacent to node } i \\ (T_{m_{im}}(r), T_{\nu_{im}}(r)) \in P_i \end{cases} \\ \\ 0 \text{ otherwise} \end{cases}$$

$T_{m_{im}}(r)$ and $T_{\nu_{im}}(r))$ are the total mean transit time and total variance in transit time respectively of the rth ant optimising mean transit time and travelling from node i to the destination node. $(T_{m_{im}}(r), T_{\nu_{im}}(r)) \in P_i$ denotes that the rth route (minimising mean transit time) from node i is on the Pareto front. A similar equation (to (14)) then gives the variance in transit time pheromone update.

5.4 Comment

The technique has a dynamic-programming basis [17] in which initially simple problems are considered and through an iterative process complete solutions to more challenging scenarios are built. The basis of the technique is to allow ants to travel from every single node in the network to the destination node, but only the first edge the ant takes being updated by its pheromone.

By building a Pareto front of solutions from each node in the network to the destination node it is the first decision of which node to go to that is of critical importance, because the solution space at/from that node is being simultaneously built. At nodes close to the destination node the number of potential routes decreases and more complete solution spaces (Pareto fronts) will exist. Iterating backwards, applying the fundamental principles of dynamic programming, we can build complete solution spaces for routing between each node in the network and the final destination.

We would expect the added complexity of this new (hybrid) technique to result in a huge increase in computation time/complexity when compared to the standard ant algorithm (of section 4). However, in updating the two pheromone matrices only the first edge selected by the ant is updated, and as a result it will be shown that the speed of the new technique far exceeds that of its predecessor.

6 Results

We shall now compare the capability of the two algorithms (the hybrid ant algorithm and the standard bi-criterion ant algorithm) to find high quality solutions. The two techniques were tested on randomly created networks of various sizes from 25 to 250 nodes with each node connected to its six nearest neighbors.

Figure 2(a) gives the lowest mean transit time determined under each algorithm for a range of networks of different sizes. Figure 2(b) gives the lowest variance in transit time in each case. These represent the two extreme edges of the Pareto front. We observe that in both cases the new technique outperforms the standard ant algorithm, the margin increasing with the size of the network.

These results may suggest that the new technique concentrates solely on finding the optimal mean transit time route and the optimal variance in transit time route, in which case an approach based solely on dynamic programming would be much better/quicker. Moreover, in comparing the two algorithms, it is insufficient simply to analyze each of the two objectives in isolation, because for the bi-criterion problem it is the combination of the mean and variance of each route that is important.

However, figure 3 clearly demonstrates that the new technique maintains a large number of non-dominated routes both at the start node (figure 3(a)) and at all nodes (figure 3(b)) across the network. We note that the average number of solutions on the Pareto front is lower when we average across the whole network because nodes near or adjacent to the destination node will obviously have few non-dominated solutions.

Furthermore, the reduced computational complexity of the new technique (hybrid) when compared to the standard technique is clearly demonstrated in figure 4(a). It can be seen that the hybrid technique typically finds the optimal solution in less than 15 seconds, whereas the standard technique did not find an optimal solution in less than 1 minute for any of the problem instances considered.

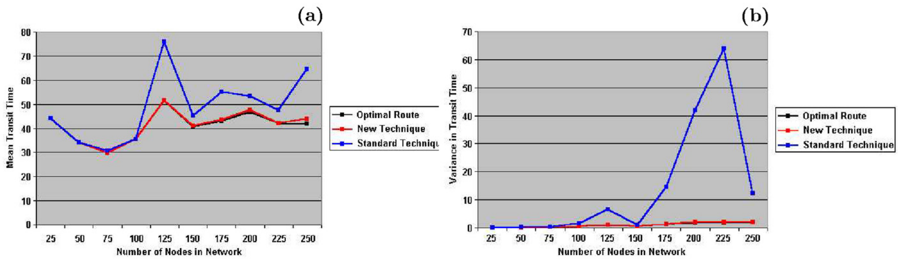

Fig. 2. **(a):** The lowest mean transit time, and **(b):** the lowest variance in transit time, for different network sizes.

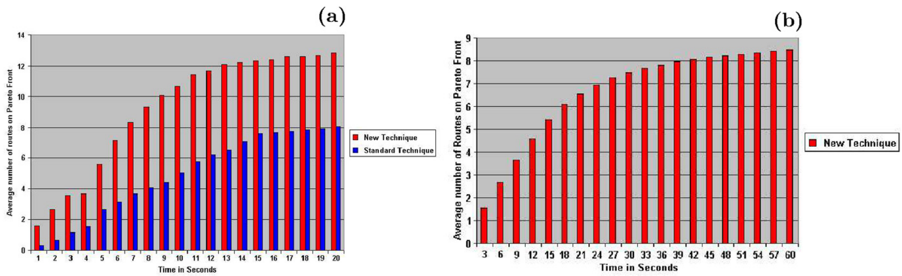

Fig. 3. **(a):** Number of non-dominated routes found by the algorithm at the start node, **(b):** number of non-dominated routes found, averaged over all the nodes in the network. In each case results are averaged over 150 runs on 100 node network problems.

With a set of non-dominated solutions held at every node the system is now also allowed to be dynamic. For example, consider a vehicle that is following the lowest mean transit route. If enroute it finds its path blocked it will have to divert. Using the new technique a new route is easily found by examining all the adjacent nodes for the route that most readily satisfies the operational requirements.

This is demonstrated by figure 4(b). The minimum mean transit time route (from the start node to the destination node) is shown by the green line. At several points the vehicle following the green route finds its path blocked by an obstacle (blue nodes) which was not there when the original route was determined. With this new technique, information is held at all the nodes in the network. Hence the adjacent nodes can be examined and the best route (here in terms of minimum mean transit time) across all the adjacent nodes is selected. This alternative route is shown by the red lines in figure 4(b).

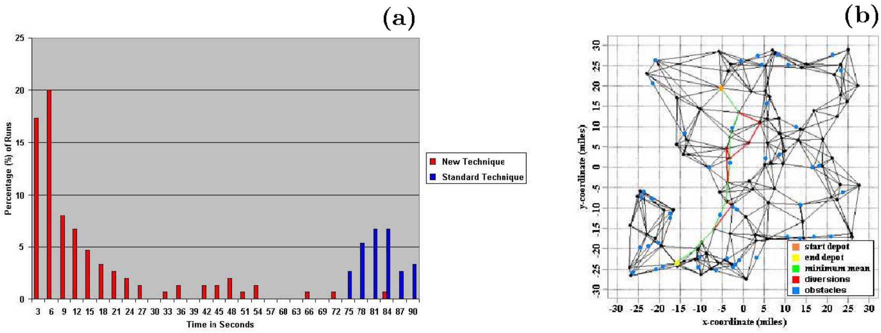

Fig. 4. **(a):** Time taken by the new ant algorithm to find the best routes in terms of mean transit time and variance in mean transit time. Results are averaged over 150 runs on 100 node network problems. **(b):** Route of minimum mean with diversions to avoid blocked vertices.

7 Conclusions

The hybrid ant algorithm presented in this report has been shown to be successful in solving bi-criterion Vehicle Routing Problems. The hybrid technique comprehensively outperformed the standard algorithm by:

- always finding the lowest mean transit and lowest variance transit time routes;
- finding considerably more routes along the Pareto front;
- running in much quicker time;
- finding a Pareto front of solutions at every single node, allowing instantaneous re-routing subsequent to network change.

Acknowledgments. This research was sponsored by the United Kingdom Ministry of Defence Corporate Research Programme CISP.

References

1. Deb, K. (1999),“Multi-Objective Optimization using Evolutionary Algorithms”, Wiley.
2. Beasley, J. E. and Christofides, N. (1989), “An Algorithm for the Resource Constrained Shortest Path Problem” Networks, 19, pp 379-394.
3. Dorigo, M., Maniezzo, V. and Colorni, A. (1991), “The Ant System: An Autocatalytic Optimizing Process”, Technical Report 91-016 (revised), Politecnico di Milano, Italy.

4. Colorni, A., Dorigo, M. and Maniezzo, V. (1991), "Distributed Optimization by Ant Colonies", Proceedings of the European Conference on Artificial Life, eds. Varela, F. and Bourgine, P. Elsevier.
5. Denebourg, J. L., Pasteels, J. M. and Verhaeghe, J. C. (1983), "Probabilistic Behaviour in Ants: a Strategy of Errors?", Journal of Theoretical Biology, 105, pp 259-271.
6. Costa, D. and Hertz, A. (1997), "Ants can colour graphs", Journal of the Operational Research Society, 48, pp 295.
7. Bullnheimer, B., Kotsis, G. and Strauss, C. (1997), "Applying the Ant System to the Vehicle Routing Problem", Proceedings of the Second International Conference on Metaheuristics.
8. Bullnheimer, B., Hartl, R. F. and Strauss, C. (1999), "An improved ant system algorithm for the vehicle routing problem", Annals of Operations Research, 89, pp 319-328.
9. Schoonderwoerd, R., Holland, O. and Bruten, J. (1996), "Ants for load balancing in telecommunications networks", Hewlett Packard Laboratory Technical Report.
10. Di Caro, G. and Dorigo, M. (1998), "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artificial Intelligence Research, 9, pp 317-365.
11. Guntsch, M., Middendorf, M. and Schmeck, H. (2001), "An Ant Colony Optimization Approach to Dynamic TSP", Proceeedings of the Genetic and Evolutionary Computation Conference, pp 860-867.
12. Jungnickel, D. (1998), "Graphs, Networks and Algorithms", Springer-Verlag.
13. Iredi, S., Merkle, D. and Middendorf, M. (2001), "Bi-Criterion Optimization with Multi Colony Ant Algorithms", Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, ed. E. Zitzler, Springer-Verlag.
14. Maniezzo, V., Colorni, A. and Dorigo, M. (1994), "The Ant System Applied to the Quadratic Assignment Problem", Technical Report IRIDIA/94-28, Universite Libre de Bruxelles.
15. Colorni, A., Dorigo, M., Maniezzo, V. and Trubian, M. (1994), "Ant System for Job Shop Scheduling", Belgian Journal of Operations Research, Statistics, and Computer Science, 34, pp 39.
16. Parpinelli, R. S., Lopes, H. S. and Freitas, A. A. (2001), "An Ant Colony Based System for Data Mining: Applications to Medical Data", Proceeedings of the Genetic and Evolutionary Computation Conference, pp 791-797.
17. Bellman, R.E. (1957), "Dynamic Programming", Princeton University Press.