An Ant System Algorithm for the Mixed Vehicle Routing Problem with Backhauls

A. Wade (wadeac@for.mat.bham.ac.uk) and S. Salhi
(s.salhi@bham.ac.uk)
Management Mathematics Group
School of Mathematics and Statistics
The University of Birmingham, U.K.

September 10, 2002

Abstract. Ant system algorithms have been used successfully to solve many hard combinatorial problems. In this paper we introduce an ant system method to solve the mixed vehicle routing problem with backhauls. Some enhancements to the general characteristics of ant system algorithms are proposed. We concentrate on the way the candidate list is constructed, a look ahead scheme to incorporate into the visibility function and efficient rules to deal with local as well as global updating. Computational results on test problems are reported and compared to known results.

Keywords: metaheuristic, ant system, vehicle routing, backhauls

1. Introduction

The ant system is a metaheuristic developed for the solution of hard combinatorial optimization problems. It was first proposed by Colorni et al. (1991) and Dorigo (1992) and used to solve the travelling salesman problem (TSP). The method was inspired by the observation of the behaviour of real life ant colonies, in particular the way in which real ants find the shortest path between food sources and their nest. While walking ants deposit a substance called pheromone on to the ground which forms a pheromone trail. Ants can detect the pheromone and choose their way according to the level of the pheromone trail. The greater the concentration of pheromone on the ground the higher the probability that an ant will choose that path. Where there is a shorter path from a food source to the nest, ants will reach the end of the path in a quicker time compared to ants on a longer path. This means that the trail will build up at a faster rate on the shorter path which in turn causes more ants to choose the shorter path which also causes a greater level of pheromone. In time all ants will have the tendency to choose the shorter path.

This real life behaviour of ants has been adapted to solve combinatorial optimization problems using simulation. A number of artificial ants build solutions in parallel using a form of indirect communication. The artificial ants co-operate via the artificial pheromone level deposited on arcs which is calculated as a function of the quality of the solution found. The amount of pheromone an ant deposits is proportional to the quality of the solution generated by that ant helping direct the search towards good solutions. The artificial ants construct solutions iteratively by adding a new node to a partial solution using information gained from both past performance and a greedy heuristic. The greedy heuristic, known as the visibility, is introduced in an attempt to guide the search.

Ant systems have been used successfully to solve many hard combinatorial problems such as the travelling salesman problem (see Dorigo et al. (1996), Dorigo and Gambardella (1997), Stutzle and Hoos (1997a) and Cordon et al. (2000)), the quadratic assignment problem (see Maniezzo et al. (1994) and Maniezzo and Colorni (1999)), the sequential ordering problem (see Gambardella and Dorigo (1997)) and the vehicle routing problem (see Bullnheimer et al. (1997), (1998) and Gambardella et al. (1999)).

The vehicle routing problem (VRP) is known to be NP-hard in the strong sense (see Garey and Johnson (1979)). This means that finding a solution to a problem which is an extension of the VRP, such as the VRP with backhauls (VRPB), is NP-complete. Therefore exact methods are only suitable to solve relatively small instances of vehicle routing problems. In these circumstances intelligent heuristic search methods are the best way forward to address such complex combinatorial problems. In this paper, we propose some enhancements to the ant system metaheuristic to solve one variant of the VRPB.

First, in the remainder of this section, we introduce the VRPB. In Section 2 we present the fundamentals of ant system algorithms and introduce successful ant system methods used to solve simpler versions of the VRPB namely the travelling salesman problem and the vehicle routing problem. Section 3 describes our basic ant system to solve the VRPB and also our modifications to the general components of ant system methods. Computational results are reported in Section 4 and finally conclusions and ideas for future research are presented in Section 5.

1.1. The Vehicle Routing Problem with Backhauls

The vehicle routing problem (VRP) consists of a number of homogeneous vehicles based at a single depot and a number of customers to be served. Each customer has a demand quantity to be delivered from the depot and a service time allocated. The objective is to find a set of routes with the least cost where all customer demands are met, every customer is visited exactly once, the total demand on each vehicle does not exceed the maximum vehicle capacity, the total length of each route does not exceed the maximum bound and every vehicle starts from and ends at the depot.

The vehicle routing problem with backhauls is an extension to the vehicle routing problem (VRP) where two different types of customer are served. The first type are linehaul customers, also known as delivery customers, who require a given quantity of product to be delivered. The second type are backhaul customers, also known as pickup customers, who require a quantity of goods to be picked up and sent back to the depot. The goods are transported to or from a single depot by a fleet of homogeneous vehicles. Methods in the literature to solve the VRPB either restrict all backhauls to be visited once all linehauls have been served, the classical VRPB, or allow mixed linehaul and backhaul customers along the routes, the mixed VRPB. In this paper we propose an ant system to solve the mixed VRPB. The objective is to find the set of routes with the least cost such that (i) all customer demands are satisfied, (ii) each customer is visited exactly once, (iii) all routes begin and end at the depot, and finally (iv) the load on the vehicle at any point along the route must not exceed the vehicle capacity.

The mixed problem is more complicated than the classical VRPB due to the fluctuating load on the vehicle. In the classical VRPB it is only necessary to check that the total linehaul load and the total backhaul load do not separately exceed the total vehicle capacity. This is also known as the VRP with pickups and deliveries (VRPPD) where all deliveries are performed first then pickups second. Gendreau, Laporte and Vigo (1999) investigate the TSP with pickups and deliveries, and Osman and Wassan (2002) put forward a useful implementation of tabu search to solve the VRP with backhauling. For more details and references on the classical VRPB or VRPPD, see Toth and Vigo (1996), Toth and Vigo (1997), Goetschalckx and Jacobs-Blecha (1989) and Mingozzi et. al. (1999). In the mixed VRPB, which is also referred to as the mixed VRPPD, it is necessary to check that the vehicle capacity is not exceeded at any point along the route. In the mixed VRPB the vehicle load can either decrease or increase at each customer depending on whether the customer has a linehaul or a backhaul demand, respectively. The mixed VRPB has not received as much attention in the literature as its counterpart the classical VRPB. Deif and Bodin (1984) use a savings based heuristic based on the Clarke and Wright algorithm to solve the mixed VRPB with a penalty function to delay the insertion of a backhaul in a route. Casco et al. (1988) propose an insertion based heuristic method which uses a penalty function based on delivery load after pickup. The aim of the penalty function is to encourage backhaul customers to be serviced towards the end of the routes. Salhi and Nagy (1999) develop an insertion based heuristic which considers a cluster of backhauls for insertion on to linehaul routes at each iteration. Halse (1992) adopts a cluster-first route-second heuristic using a relaxed assignment problem and modified route improvement procedures to solve the mixed VRPB. In this study we do not address the simultaneous VRPPD where customers may simultaneously receive and send goods.

The main reason for imposing the restriction that backhauls can only be served after all linehauls have been visited on a route, apart from the fact that the mixed problem is more complicated to solve, is that many of the routes that are created if mixed customers are allowed may not be practical to implement. This is due to the fact that it may be necessary to remove backhaul load from the vehicle at subsequent visits to linehaul customers. However, we feel that in many real life applications it may not be necessary to restrict all backhauls to the end of the routes. In such cases significant savings can be achieved. The improved design of vehicles, in particular vehicles that permit side loading, means that mixed routes are now a much more practical option and therefore worth investigating.

2. Ant Colony Optimisation (ACO)

2.1. A brief overview on ACO

The Ant System algorithm (AS), first proposed by Dorigo (1992) and Colorni et al. (1991) to solve the travelling salesman problem uses artificial pheromone trail values, τ_{ij}, associated with each arc (i, j). Initially m ants are placed on randomly selected nodes and each ant starts constructing a tour from this starting position. The tour is built by each ant successively choosing the next customer to visit probabilistically. The probability that an ant building its t^{th} tour currently situated at customer i will visit customer j next is given by the state transition rule:

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in F_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} \qquad if \quad j \in F_i^k, \quad 0 \quad \text{otherwise} \qquad (1)$$

where η_{ij} is a local heuristic function, known as the visibility, usually given as $\eta_{ij} = 1/d_{ij}$. d_{ij} is the distance from customer i to customer j, and F_i^k is the list of feasible customers not yet visited by ant k. The parameters α and β determine the relative influence of the trail value and the heuristic information. To avoid the use of two parameters, one can set say $\alpha = 1$ and choose the value of β accordingly.

The ants keep track of the customers that have been visited in the order they were served to form a partial tour. Once all ants have constructed a complete tour, i.e. all customers have been visited, then the pheromone trail levels are updated by each ant according to the global update rule given in the equation

$$\tau_{ij}(t) \longleftarrow (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{2}$$

where

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L_k(t) & if \quad (i,j) \in T^k(t) \\ 0 & \text{otherwise} \end{cases}$$

and $0 < \rho \leq 1$ is the pheromone trail evaporation. The parameter ρ is used to avoid stagnation in the solution which would occur if the trail values were unrestricted, Q is a correction factor used to normalize the data set, $L^k(t)$ is the length of the tour produced by ant k at iteration t and $T^k(t)$ is the tour produced by ant k at iteration t.

A number of modifications have been proposed to the AS algorithm in order to improve the solution quality. The elitist strategy which was proposed by Dorigo et al. (1996) is based on a modified global updating rule where the arcs belonging to the global best tour received an extra amount of pheromone. This idea is then generalised by Bullnheimer et al. (1997) to include not only the global best tour but also a number of the best tours to update trail values. The ants are ranked according to their tour length and the amount of pheromone an ant adds is proportional to its rank. The Ant Colony System (ACS) proposed by Dorigo and Gambardella (1997) include three main modifications: a different transition rule based on the pseudo-random-proportional rule, a pheromone global updating rule based only on the global best solution, and a local updating rule that adjusts the pheromone level on the selected arcs. The ACS method makes use of a candidate list to solve large combinatorial problems. The candidate list is used to reduce the number of choices available to each ant when the ant is considering its next move, hence reducing the computation time. The candidate list is one of the key factors which we will investigate in this study. Stutzle and Hoos (1997a), (1997b) introduced the Max-Min ant system, known as MMAS. Their method is basically similar to AS except that pheromone trail values are restricted to the interval $[\tau_{min}, \tau_{max}]$ to avoid stagnation. The introduction of a lower limit on pheromone trail values was found to have the greatest effect on the solution quality.

For an overview on ant systems the book of Bonabaeu et al. (1999), the review paper by Dorigo et al. (1999) and the technical report by Dorigo and Stutzle (2000) are useful references.

2.2. Ant Systems for Vehicle Routing Problems

Bullnheimer et al. (1997) (1998) propose an ant system to solve the VRP. Their method, named AS-VRP, is based upon their AS_{Rank} method. Bullnheimer et al. (1997) extend the visibility function to be problem specific by incorporating a savings function and capacity utilization. The revised visibility function produced better results but was found to be costly in respect of computation time. In (1998) a parametric savings function is adopted for the visibility producing an improvement in solution quality and a reduction in computation time. A 2 opt local search procedure is implemented to improve solutions. Bullnheimer et al. (1997) (1998) adapted the list of feasible customers for each ant so that the capacity constraint and the maximum route length are taken into account. If no feasible customers are available to an ant building its tour then the ant returns to the depot so that a new route can be started.

Gambardella et al. (1999) also solve the VRP using an ant based algorithm. The problem is reformulated by adding $(M - 1)$ depots, where M is the number of vehicles. The problem therefore becomes a travelling salesman problem with added constraints. The algorithm Gambardella et al. develop, called HAS-VRP, is based upon the ACS method. Each ant builds its tour so that the vehicle capacity is not exceeded. Each tour that is generated contains M subtours each corresponding to the route of a vehicle. A local search heuristic comprising edge exchanges is applied to the solutions. Gambardella et al. also solve the vehicle routing problem with time windows (VRPTW) using an ant algorithm. The VRPTW is an extension to the VRP where each customer must be serviced within a given time interval. Two ant colonies are utilized by their method. The first ant colony is used to minimize the number of vehicles and the second colony is used to minimize the total travel time given the number of vehicles determined by the first ant colony. The pheromone trail levels are independent for each colony but the best ants from one colony are allowed to update trail values associated with the other colony.

The results reported by the methods detailed in this subsection have been shown to be competitive when compared with known methods

```
Initialize
For I=1 to number of iterations
    For m =1 to number of ants
        For n=1 to number of customers
            Select the next customer to visit using state
            transition rule
            Update the pheromone trail for the selected arc
            using local updating rule
        End for
        Apply fast local search heuristics
    End for
    Update pheromone trail values using global updating rule
End for
```

Figure 1. Ant System Algorithm for VRPB

in the literature to solve the vehicle routing problem and the vehicle routing problem with time windows.

3. An Ant System Heuristic for the VRPB

We propose an algorithm based upon the Ant Colony System, used successfully to solve the TSP, for the VRPB. A number of modifications to the ACS method are proposed but the basic ant algorithm can be described as follows. Initially m ants are positioned at starting customers. Each ant selects the next customer to visit according to the state transition rule which will be looked at in this section. Once an arc (i, j) has been selected by an ant then a local updating rule is applied to reduce the trail value of that arc. Each ant constructs its tour selecting one customer at a time. If no feasible customers are available to be visited due to the vehicle capacity constraint then the depot is selected by the ant. The ant then continues to build a new route repeating the process until all customers have been selected. Two local search procedures are adopted accordingly for the VRPB. A restricted 2/3 opt procedure is used to improve each solution after each ant has completed its tour. A shift procedure that exchanges customers between routes is also introduced. The structure of our ant algorithm is given in Figure 1.

The modifications we propose to the Ant Colony System are detailed in the remainder of this section.

3.1. A Site Dependent Candidate List

Many ant system procedures use a data structure known as a candidate list. A candidate list contains for each customer, i, a given number of potential customers to be visited. These favoured customers can be chosen according to their distance from customer i. A candidate list is a static data structure containing a predetermined number of customers. An ant first chooses the next customer to be visited from the candidate list corresponding to the current customer. Only if it is not feasible to visit any of the customers belonging to the list then the ant selects the next customer to visit as the closest out of all feasible customers and the depot. We adopt this procedure in our algorithm unless there is only one remaining customer to be visited. In our method if only one customer remains to be visited in the problem and the customer is contained in the feasible list but not the candidate list, then that customer is forced to be visited next. This should avoid one remaining feasible customer being forced into a new route if the depot had been the closest option.

The purpose of the candidate list is to reduce the possible number of moves that an ant must evaluate. This is performed by recording for each customer only those who are not considered far away customers to that particular customer. In other words, the path between the current customer and one of those far away customers is unlikely to be chosen as such a link is unlikely to produce a good quality solution. Looking at Figure 2, if we consider customer A it may not be sensible for an ant currently at customer A to consider customer B as the next move in view of solution quality.

It can be observed that using a static candidate list does not take into consideration the number of customers situated in the surrounding neighbourhood of customer i. For instance in Figure 2, customer A has only a few customers in the near vicinity. Therefore, if the size of the candidate list is set too high then far away customers may be considered which would not be a sensible choice. If we now consider customer B, which has many customers in the surrounding neighbourhood, then if the number in the candidate list is set too low customers which may lead to a good solution could be ruled out. We put forward an approach that is site dependent, i.e. that takes into account the sparsity of customers in the surrounding neighbourhood of each customer.

The minimum number, M, contained in the candidate list for each customer is given by:

$$M = Min\{N/4, 10\}$$

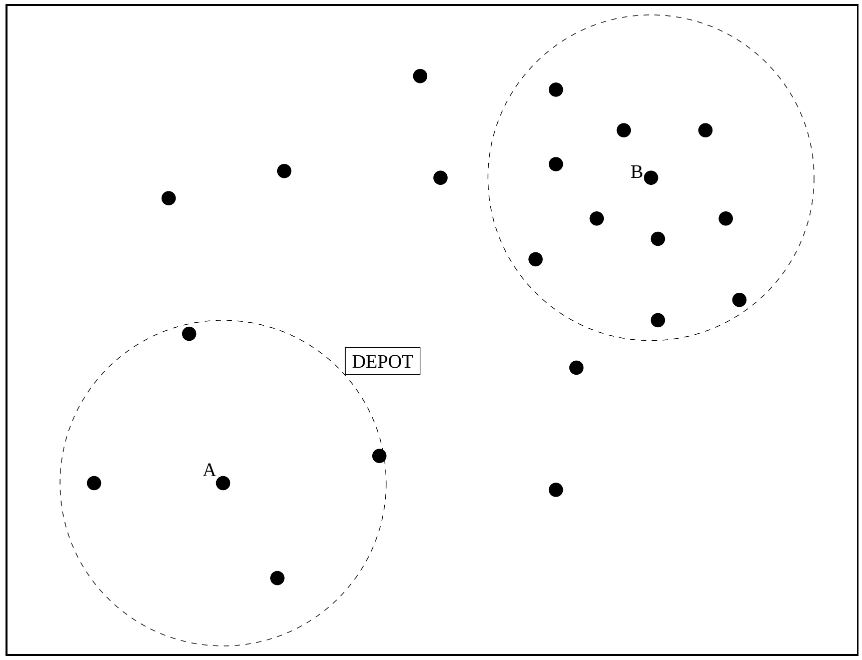

Figure 2. Example Problem

where N is the total number of customers. The distance from customer i to all other customers, d_{ij} is calculated and the average of these distances is found as follows:

$$\tilde{d}_i = \frac{\sum_{j \neq i} d_{ij}}{N-1}$$

This process is repeated for all customers and the average of these results is given by:

$$D = \frac{\sum_{k=1}^{N} \tilde{d}_k}{N}$$

For each customer the distance to all other customers that fall within the range D is calculated and the average of all these distances is computed:

$$C_i = \frac{\sum_{\{j : d_{ij} < D\}} d_{ij}}{|\{j : d_{ij} < D\}|}$$

In addition, if $|\{j : d_{ij} < D\}| = 0$ set $C_i = 0$.
The overall average is found by:

$$R = \frac{\sum_{k=1}^{N} C_i}{N}$$

The candidate list is then constructed for each customer i as $E_i = \{j = 1, \ldots, m \quad s.t. \quad d_{ij} \leq R\}$. In addition if $|E_i| < M$ then the nearest $M - |E_i|$ customers not already contained in E_i are selected, say O_i, then we set $E_i = E_i \cup O_i$. In particular if E_i is empty, we set $E_i = O_i$.

3.2. Strategic Placement of Ants

In the majority of ant algorithms it is common to place one ant on each node. In this paper, we relax such a restriction and we adopt a more strategic approach to the placement of ants.

The whole region originated from the depot is divided into a given number of sectors which is based on the number of customers. The sectors are constructed so that each contains an equal number of customers where possible. If this is not possible then the remaining customers are allocated to the final sector. Within each sector an ant is placed at the closest and farthest customer in relation to the depot. A given number of ants are then placed randomly on nodes remaining in each sector. For each iteration of the ant algorithm the starting position in the calculation of the sectors is rotated. This procedure should result in an even spread of ants being selected for each iteration. The aim of this procedure is that the sectors should roughly correspond to the routes that would be created in a good solution. The restriction on the number of ants can be altered within the algorithm so that all customer sites can still be selected if desired. In our algorithm we use $N/2$ ants, where N is the total number of linehauls and backhauls. The restricted number of ants has the effect that fewer solutions are created which could have a detrimental effect on solution quality. However, computation time decreases which means that more iterations can be performed than would be the case if the number of ants is set equal to the number of customers as used in most applications. Because more iterations are performed to generate a given number of solutions, the ants can take advantage of a greater number of pheromone trail updates hence learning from past experience at a faster rate.

3.3. A Look Ahead based Visibility

As discussed in Section 2 the visibility is a greedy heuristic used within the transition rule in an attempt to guide the search. For the travelling salesman problem the greedy heuristic is generally calculated as the

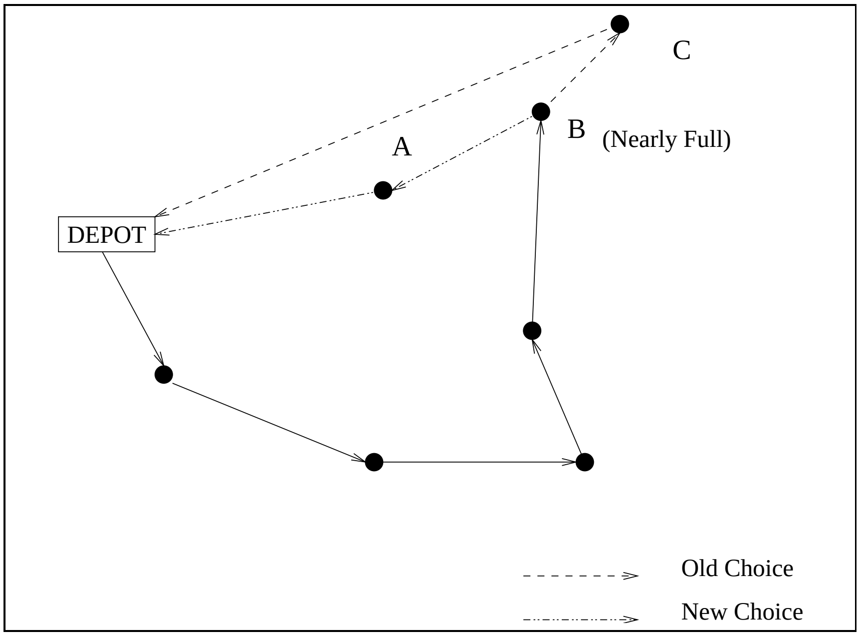

Figure 3. Insertion Based Visibility

reciprocal of the distance between the current customer where the ant is placed and the customer to which the ant is considering to move. Due to the fact that the visibility function is used to guide the search, it can have a significant influence on the quality of results. In this paper we propose a visibility that takes into account the load on the vehicle in the selection process. Two different visibility functions are used, the choice of which is dependent on the capacity remaining on the vehicle.

(i) If the vehicle is nearly full so that the vehicle is near to the end of its route then it would not be a sensible choice to visit a customer which is further away from the depot than the current customer. It would be more efficient to visit a customer that is between the current customer and the depot if possible. This can be seen in Figure 3 where the vehicle is currently positioned at customer B. The load on the vehicle at customer B is near the maximum capacity constraint and can only accommodate the load of A or C but not both. Therefore it would not be sensible to visit customer C next as there may not be enough room remaining on the vehicle to service another customer but the vehicle has travelled further away from the depot. It is more likely to be an efficient choice to visit customer A after customer B due to the fact that the cost of inserting customer A between B and the depot is relatively smaller.

The visibility we propose uses the insertion cost between the current customer and the depot in order to encourage the vehicle to travel closer to the depot towards the end of each route. If the unused capacity on the vehicle is less than a given parameter based on the average linehaul demand then the visibility is calculated as:

$$\eta_{ij} = \frac{1}{d_{ij} + d_{j0}}$$

where j is the customer to which the ant is considering to move and customer 0 is the depot.

(ii) At the beginning of routes, before the above cut off point is reached, we propose a visibility that takes into account a chain of two customers. When considering the next customer j to be visited then the nearest customer k in relation to j is incorporated into the visibility detailed in Equation (3). Note that in previous applications κ_{il} is not included so if $\phi = 0$ our transition rule reduces to the one given in ACS by Dorigo and Gambardella (1997).

$$j = \begin{cases} argmax_{u \in F_i^k} \{[\tau_{iu}(t)]^{\alpha} \cdot [\eta_{iu}]^{\beta} \cdot [\kappa_{iu}]^{\phi}\} & \text{if } q \leq q_0 \\ S & \text{if } q > q_0 \end{cases} \tag{3}$$

where
q is a random variable uniformly distributed in the interval [0,1], q_0 is a parameter ($0 \leq q_0 \leq 1$) and S is a random variable selected according to the following probability distribution

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta} \cdot [\kappa_{ij}]^{\phi}}{\sum_{l \in F_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta} \cdot [\kappa_{il}]^{\phi}}$$

$$\kappa_{ij} = \frac{1}{d_{ij} + d_{jk}}$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

The idea is to incorporate such a look ahead strategy that takes into consideration the next possible move following the selection of customer j. In our implementation we do not check for feasibility when considering customer k in the chain due to the significant increase in computation time which would be required. Therefore for each customer, i, the value of κ_{ij} is fixed for all iterations irrespective of whether it would be feasible to visit customer k later along the route. It should also be noted that η_{ij} is also incorporated in to the visibility so that the probability function is not solely influenced by the chain of customers i, j, k.

3.4. A Frequency based Local Trail Update

The local trail update as defined by Dorigo and Gambardella (1997) is used to encourage exploration of the search space. In other words, when an ant has selected the next customer to visit the trail value along the corresponding arc is reduced. To promote exploration within the search, one way is to encourage successive ants to select other arcs with a greater probability. In some cases the trail value on an arc may be very high or may be dominated by the visibility in which case reducing the trail value may not be enough to encourage the selection of other arcs. We propose a procedure that records the number of times each arc has been selected. If an arc has been selected a greater number of times than a given percentage, *perc*, of the number of solutions that have been generated since the beginning of the algorithm, then the trail value on that arc is updated according to Equation (4), otherwise we apply the same equation but with $\gamma = 1$. The latter becomes the local trail update rule used in ACS by Dorigo and Gambardella (1997).

$$\tau_{ij} \longleftarrow (1 - \gamma\upsilon) \cdot \tau_{ij} + \upsilon \cdot \tau_0 \tag{4}$$

where υ is a pheromone decay parameter in the range $(0 < \upsilon < 1)$, γ is our adjustment factor which we introduced $(\gamma > 1)$, and τ_0 refers to the initial pheromone trail level. The value of τ_0 is usually set to $\frac{N}{Cost(H)}$ where $Cost(H)$ refers to the cost found by a given heuristic. In this study we approximate such a value based on our preliminary results on cost.

The effect is to reduce the pheromone trail value by a greater amount on arcs that have been selected many times compared to arcs that have only been picked by a few ants. This procedure is designed to reduce the possibility that arcs are selected repeatedly and to encourage exploration of the search space. It is also intended that customers which are chosen towards the beginning of a route by one ant will be selected later along a route by other ants so that mixing occurs on routes. This procedure is only implemented after a limited number of complete iterations of the algorithm have been performed. For instance, in our experiments we opted for three complete iterations based on our limited computational experience. The aim is to allow the ants to generate routes freely according to the trail values before any influence is applied.

3.5. Combining Strategies for Global Trail Update

In our method we use a global trail update based on the rank based version with elitist strategies, AS_{rank}, proposed by Bullnheimer et al. (1997). In their method the global best solution found so far is used to update the trail values together with a given number, λ, iteration best solutions. The ranking system means that the trail value update is weighted according to the rank of the solution. However, this method does not take into account the quality of the λ iteration best solutions created when compared to the global best solution. This means that if λ is set too small then good quality solutions may not be used to update trail values if many good solutions have been generated. Also if λ is set too large then bad solutions may be used to update trail values if only a small number of good quality solutions have been built in the current iteration.

We propose a method that does consider the quality of the solutions generated. The aim is to reduce the risk that the number of ants used to update trail values has been set too high if we want to make sure that many good solutions are used in the updating process where appropriate. Our procedure uses the global best solution to update pheromone trail values together with a maximum number, λ, of iteration best solutions. Considering each of the λ iteration best solutions in turn, only if the solution is within a given percentage, θ, of the global best solution is the route used to update trail values. The trail values are updated according to Equation (6) as given by Bullnheimer et al. (1997) in their AS_{rank} except that λ may be adjusted as follows:

Let the set of the best λ solutions whose cost lie within $\theta\%$ of the global best solution be defined by Equation (5).

$$G = \{s = 1, \ldots, \lambda \text{ s.t. } \frac{(cost(s) - cost(best))}{cost(best)} \cdot 100 < \theta\}. \tag{5}$$

If $|G| < \lambda$ we set $\lambda = |G|$.

The constraint on cost is imposed to avoid poor quality solutions being used to update the trail values.

$$\tau_{ij}(t) \longleftarrow (1 - \rho) \cdot \tau_{ij}(t) + \lambda \Delta \tau_{ij}^{gb} + \Delta \tau_{ij}(t) \tag{6}$$

where $\Delta \tau_{ij}^{gb}(t) = Q/L^{gb}(t)$, L^{gb} being the global best solution and Q a correction factor to normalise the data.

$\Delta \tau_{ij}(t) = \sum_{\mu=1}^{\lambda-1} \Delta \tau_{ij}^{\mu}(t)$.

$$\Delta\tau_{ij}^{\mu}(t) = \begin{cases} (\lambda-\mu)Q/L^{\mu}(t) & \text{if } (i,j) \in \text{tour of ant with rank } \mu \\ 0 & \text{otherwise} \end{cases}$$

$L^{\mu}(t)$ is the length of the μ^{th} best tour at iteration t.

3.6. Simple post-optimisers

To control the quality of the solutions obtained we adopted into our ant heuristic two simple and well known local search procedures namely the 2/3 opt approach and the shift heuristic. In the 2/3 opt procedure only customers which belong to the same route are considered and hence each route is improved independently. A local search algorithm that works on the entire set of routes is obviously more powerful but can be expensive. In this work we adopted a route improvement heuristic which exchanges one or two customers between routes. These refinement procedures, which are used by Salhi and Rand (1987) for the vehicle routing problem, are modified for the vehicle routing problem with backhauls.

Note that it is not possible to apply standard post-optimisers because of the fluctuating load on the vehicle caused by the two different types of customer. In other words, exchanging a linehaul or backhaul customer in the VRPB can affect the load on the vehicle and could mean that the vehicle capacity is exceeded at other nodes along the route. It should also be noted that due to the great number of solutions that are generated it would be time prohibitive to employ a more sophisticated local search heuristic for each solution. In our implementation we used the 2/3 opt after each iteration and for all the routes generated whereas the shift procedure is used at particular times such as when the cost of a new solution is found to be very different from the previous ones.

4. Computational Results

The set of test problems corresponds to the VRPB instances proposed by Goetschalckx and Jacobs-Blecha (1989). The problems range in size from 25 to 150 customers. The cost of travelling from customer i to customer j is defined as the Euclidean distance between nodes i and j. The ant algorithm was implemented in Fortran90 and the experiments executed on an Ultra Enterprise 450 (300 MHz dual processor). For each test problem the ant system was run for 200 iterations. The parameters were set to the following values:

Transition rule (Equation (3)): $\alpha = 1, \beta = 2, \phi = 1, q_0 = 0.5$

Local trail update rule (Equation (4)): $v = 0.4, \gamma = 2, perc = 25\%, \tau_0 = 5.10^{-5}$.

Cost constraint (Equation (5)): $\theta = 10\%$

Global update rule (Equation (6)): $\rho = 0.1, Q = 1$, $\lambda = 10$ at the first iteration, then $\lambda = max(10, NR)$ in subsequent iterations. NR is the number of routes found in the global best solution so far.

These values were chosen experimentally based on our limited preliminary testing and also on data from the literature whenever appropiate. To our knowledge this set of problems has only been solved for the mixed VRPB by Halse (1992). As the placement of the ants is randomly chosen within the sectors we conducted experiments using 20 runs for each test problem. For each problem we report the best solution, the average of the 20 solutions and the average computing time per run. Our summary results are given in Table I. Bold refers to a new best solution. In Table I we also provide the name of the problem, together with the total number of customers N, the vehicle capacity CAP and the number of vehicles found K by both heuristics. The percentage deviation with respect to results reported by Halse is given in the two final columns (A minus sign shows that the present method outperforms Halse and vice versa). Underline shows both the best and the worst deviation. The deviation (in %) is computed as $Deviation = \frac{(Cost_A - Cost_H)}{Cost_H} \cdot 100$, where $Cost_A$ and $Cost_H$ refer to the cost of the best solution found by our ant heuristic and Halse respectively.

It can be seen from the table that the results using our ant algorithm range from -7.44% to 2.69% when compared to the solutions generated by Halse. In other terms, our worst solution is just below 3% whereas our new best is over 7%. In addition, 25 new best solutions out of 45 are obtained. The average deviation for all sets of problems is -0.62% for our best results and our average solutions were found to be on average 0.71% worst. Our ant heuristic seems to produce robust solutions as demonstrated by the low values of the standard deviations.

A brief comment of the effect of the enhancements-

It is worth reporting that the enhancement to the candidate list gave an improvement of approximately 8% for the overall set of test problems when compared to the basic ant algorithm applied to the VRPB without any modifications. Also, the amendment to the visibility, local and global updating each showed an improvement of around 2% in the overall results for a given number of iterations. The restriction to the number of ants did not show an improvement when comparing results over the same number of iterations. However, by restricting the number of ants fewer routes are created, and therefore more iterations can be performed for the same computational effort. A small improvement of approximately 1% was achieved when comparing results over an equal number of solutions generated. Applying all modifications to the ant system algorithm produced an improvement of 14% taken as an average over the complete set of test problems. The detailled results are not reported here but can be found in the thesis of Wade (2002).

5. Conclusions

In this paper we have presented an ant system algorithm to solve the mixed vehicle routing problem with backhauls. We have analysed and modified a number of the basic attributes of ant system methods in order to solve the problem. It can be noted that our ant system produces encouraging results as it obtains 25 out of 45 new best results. The method performs well in most problems including the larger ones and proves to be robust.

The proposed study could make a strong basis for further enhancement. There are a number of ideas which can be incorporated into the proposed heuristic. (i) One would be to adapt the greedy heuristic or visibility to be more problem specific. The visibility is used in an attempt to guide the search and hence can have a signifcant influence on solution quality. A revised heuristic could be based on a savings function and incorporate information concerning linehaul and backhaul load at each customer. (ii) As larger problems require more solutions to be generated because of the relatively larger number of ants to be used, solutions may become dominated by large trail values on certain arcs. One way that this could be addressed would be to introduce maximum and minimum values for the pheromone trail as proposed by Stutzle and Hoos (1997a), (1997b).These values would only be enforced during given stages of the iteration. Though such modifications can increase computation time a further investigation would be worthwhile carrying out. (iii) As the number of parameters introduced in this heuristic is increased, contrarily to what one would envisage, it would be interesting to investigate ways of eliminating some parameters especially those that could be considered as correlated. A sensitivity analysis on some of those relevant parameters as well as a statistical investigation which explores the statistical significance of each of the enhancements may be useful to conduct as part of assessing further the performance of such a population based heuristic.

References

Bonabaeu, E., M. Dorigo, and G. Theraulaz: 1999, *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press.

Bullnheimer, B., R. Hartl, and C. Strauss: 1997, 'An Improved Ant System Algorithm for the Vehicle Routing Problem'. Technical Report POM-10/97, Institute of Management Science, University of Vienna.

Bullnheimer, B., R. Hartl, and C. Strauss: 1998, 'Applying the Ant System to the Vehicle Routing Problem'. In: I. Osman, S. Voss, S. Martello, and C. Roucairol (eds.): *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization.* Kluwer Academics, pp. 109–120.

Casco, O., B. Golden, and E. Wasil: 1988, 'Vehicle Routing with Backhauls: Models, Algorithms and Case Studies'. In: B. Golden and A. Assad (eds.): *Vehicle Routing: Methods and Studies.* Amsterdam: North Holland, pp. 127–147.

Colorni, A., M. Dorigo, and V. Maniezzo: 1991, 'Distributed Optimization by Ant Colonies'. In: F. Varela and P. Bourgine (eds.): *Proceedings of the European Conference on Atrificial Life.* pp. 134–142, Elsevier Publishing, Amsterdam.

Cordon, O., I. F. de Viana, F. Herrera, and L. Moreno: 2000, 'A New ACO Model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System'. In: M. Dorigo, M. Middendorf, and T. Stutzle (eds.): *Abstract Proceedings of ANTS2000- From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms.* pp. 22–29, Université Libre de Bruxelles, Belgium.

Deif, I. and L. Bodin: 1984, 'Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauls'. In: A. E. Kidder (ed.): *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Managemant.* pp. 75–96.

Dorigo, M.: 1992, 'Optimization, Learning and Natural Algorithms'. Ph.D. thesis, Politecnico di Milano, Italy.

Dorigo, M., G. Caro, and L. Gambardell: 1999, 'Ant Algorithms for Discrete Optimization'. *Artificial Life* **5**(3), 137–172.

Dorigo, M. and L. Gambardella: 1997, 'Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem'. *IEEE Transactions on Evolutionary Computation* **1**(1), 53–66.

Dorigo, M., V. Maniezzo, and A. Colorni: 1996, 'The Ant System: Optimization by a Colony of Cooperating Agents'. *IEEE Transactions Systems, Man, and Cybernetics- Part B* **26**(1), 29–41.

Dorigo, M. and T. Stutzle: 2000, 'The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances'. Technical Report IRIDIA-2000-32, Université Libre de Bruxelles, Belgium.

Gambardella, L. and M. Dorigo: 1997, 'HAS-SOP: An Hybrid Ant System for the Sequential Ordering Problem'. Technical Report 11-97,IDSIA, Lugano, Switzerland.

Gambardella, L., E. Taillard, and G. Agazzi: 1999, 'MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows'. In: D. Corne, M. Dorigo, and F. Glover (eds.): *New Ideas in Optimization.* McGraw Hill, London, pp. 63–76.

Garey, M. and D. Johnson: 1979, *Computers and Intractability: A Guide to the Theory of NP Completeness.* San Francisco: Freeman.

Gendreau, M., G. Laporte, and D. Vigo: 1999, 'Heuristics for the Travelling Salesman Problem with Pickup and Delivery'. *Computers and Operations Research* **26**, 699–714.

Goetschalckx, M. and C. Jacobs-Blecha: 1989, 'The Vehicle Routing Problem with Backhauls'. *European Journal of Operational Reasearch* **42**, 39–51.

Halse, K.: 1992, 'Modelling and Solving Complex Vehicle Routing Problems'. Ph.D. thesis, The Technical University of Denmark.

Maniezzo, V., A.Colorni, and M. Dorigo: 1994, 'The Ant System applied to the Quadratic Assignment Problem'. Technical Report IRIDIA/94-28, Université Libre de Bruxelles, Belgium.

Maniezzo, V. and A. Colorni: 1999, 'The Ant System applied to the Quadratic Assignment Problem'. *IEEE Trans. Knowledge and Data Engineering*.

Mingozzi, A., S. Giorgi, and R. Baldacci: 1999, 'An Exact Method for the Vehicle Routing Problem with Backhauls'. *Transportation Science* **33**(3), 315–329.

Osman, I. and N.A.Wassan: 2002, 'A reactive tabu search meta-heuristic for the vehicle routing problem with backhauling'. *Computers and Operations Research (to appear)* **5**.

Salhi, S. and G. Nagy: 1999, 'A Cluster Insertion Heuristic for the Single and Multiple Depot Vehicle Routing Problems with Backhauls'. *Journal of the Operational Research Society* **50**(10), 1034–1042.

Salhi, S. and G. Rand: 1987, 'Improvements to Vehicle Routing Heuristics'. *Journal of the Operational Research Society* **38**, 293–295.

Stützle, T. and H. Hoos: 1997a, 'Improvements on the Ant System: Introducing MAX-MIN Ant System'. In: *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*. pp. 245–249, Springer Verlag, Wien.

Stützle, T. and H. Hoos: 1997b, 'The MAX-MIN Ant System and Local Search for the Travelling Salesman Problem'. In: T. Baeck, Z. Michalewicz, and X. Yao (eds.): *Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference*. pp. 309–314, IEEE Press.

Toth, P. and D. Vigo: 1996, 'A Heuristic Algorithm for the Vehicle Routing Problem with Backhauls'. In: *Advanced Methods in Transportation Analysis*. Berlin: Springer Verlag, pp. 585–608.

Toth, P. and D. Vigo: 1997, 'An Exact Algorithm for the Vehicle Routing Problem with Backhauls'. *Transportation Science* **31**(4), 372–385.

Wade, A.: 2002, 'Constructive and Ant System heuristics for a class of VRP with Backhauls'. Ph.D. thesis, University of Birmingham.

Acknowledgments- The authors are grateful to both referees for their constructive comments that improved the presentation as well as the content of the paper.

Table I. Computational Results for the Mixed VRPB

			Ant Results				Halse Results			Deviation (in %) wrt Halse	
Problem	N	CAP	*Best*	K	*Average*	*cpu*	*Halse*	K	*cpu*	Best	Average
a1	25	1550	**223088**	8	223374	0.35	227725	8	0.18	-2.04	-1.91
a2	25	2550	169500	5	169797	0.31	169497	5	0.09	0.002	0.18
a3	25	4050	142034	3	142126	0.34	142032	3	0.05	0.001	0.06
b1	30	1600	**233001**	7	234514	0.58	233950	7	0.13	-0.41	0.24
b2	30	2600	**179258**	4	179760	0.76	182326	4	0.07	-1.68	-1.41
b3	30	4000	145702	3	145702	0.71	145699	3	0.30	0.002	0.002
c1	40	1800	**239192**	7	240126	1.65	242931	7	0.69	-1.54	-1.15
c2	40	2600	**196883**	5	197287	2.08	197276	5	0.74	-0.20	0.006
c3	40	4150	**164891**	3	165710	1.87	167663	4	0.06	-1.65	-1.16
d1	38	1700	**307110**	11	307383	1.56	307875	11	0.47	-0.25	-0.16
d3	38	2750	224196	7	224598	1.28	222195	7	0.18	0.90	1.08
e1	45	2650	**223774**	7	225927	2.30	222518	7	0.43	0.56	1.53
e2	45	4300	190559	4	191342	3.26	190048	4	0.07	0.27	0.68
e3	45	5225	**182804**	4	184621	2.42	187793	4	0.16	-2.66	-1.69
f1	60	3000	**248333**	7	251364	5.55	254977	6	0.53	-2.61	-1.42
f3	60	4400	217317	5	218818	6.61	215575	5	0.44	0.81	1.50
f4	60	5500	**200964**	4	202280	7.58	203448	5	0.41	-1.22	-0.57
g1	57	2700	**301235**	10	311057	7.64	304106	10	0.41	-0.94	2.28
g2	57	4300	235920	6	236623	4.26	235220	6	0.17	0.30	0.60
g3	57	5300	214534	5	215133	9.89	213757	5	0.22	0.36	0.64
g5	57	6400	203233	4	203777	5.32	202610	4	0.33	0.31	0.58
g6	57	8000	**189922**	3	190572	9.88	201875	4	0.33	-5.92	-5.59
h1	68	4000	241619	6	241850	9.94	235269	6	0.65	2.69	2.79
h2	68	5100	220305	5	220532	8.57	215649	5	0.18	2.15	2.26
h3	68	6100	208412	4	208693	8.80	202971	4	0.12	2.68	2.82
h5	68	7100	203193	4	203647	9.21	201896	4	0.24	0.64	0.86
i1	90	3000	**327168**	10	336501	22.78	329237	10	1.06	-0.63	2.21
i2	90	4000	**278727**	7	281864	27.78	289501	7	1.76	-3.72	-2.64
i3	90	5700	**238626**	5	241102	24.95	244782	5	3.83	-2.51	-1.50
j1	94	4400	**332471**	10	341272	28.02	337800	10	1.08	-1.58	1.03
j2	94	5600	**292698**	8	301779	32.36	298432	8	1.32	-1.92	1.12
j3	94	6600	**259243**	6	261264	32.15	280070	7	0.38	-7.44	-6.71
j4	94	8200	261066	7	278866	50.55	257895	6	0.53	1.23	8.13
k1	113	4100	**360954**	10	370134	57.71	361287	10	1.40	-0.09	2.45
k2	113	5200	323979	8	327324	57.99	320012	8	1.57	1.24	2.28
k4	113	6200	298518	7	308065	58.79	296766	7	0.66	0.59	3.81
l1	150	4000	416167	11	418644	159.97	412278	10	3.81	0.94	1.54
l2	150	5000	**360018**	8	377962	164.42	362399	8	4.01	-0.66	4.29
l4	150	6000	**337620**	7	339020	160.74	341304	7	2.09	-1.08	-0.67
m1	125	5200	**370920**	10	381006	70.79	372840	11	2.41	-0.51	2.19
m3	125	6200	**335486**	9	354797	80.21	336011	9	2.71	-0.16	5.59
m4	125	8000	310567	7	311410	83.48	305118	7	0.82	1.78	2.06
n1	150	5700	**370690**	10	387926	129.97	385978	10	2.60	-3.96	0.50
n3	150	6600	**349516**	9	349555	151.62	352992	9	2.82	-0.98	-0.97
n5	150	8500	323698	7	332754	166.68	319811	7	3.11	1.21	4.05
Overall Average										**-0.62**	0.71
Standard Deviation										2.01	2.56
Number of Best			25		14		20				