Comparing backhauling strategies in vehicle routing using Ant Colony Optimization

Marc Reimann and Heinz Ulrich

Institute for Operations Research, ETH Zurich
Clausiustrasse 47, CH-8092 Zurich, Switzerland
e-mail: Marc.Reimann@ifor.math.ethz.ch, Heinz.Ulrich@ifor.math.ethz.ch

Abstract In the Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW) customers either receive goods from the depot or send goods to the depot and pickup or delivery at a customer has to occur within a pre-specified time window. The main objective is to minimize the total required fleet size for serving all customers. Secondary objectives are to minimize the total distance travelled or to minimize the total route duration of all vehicles. In this paper we consider a variant of the mixed VRPBTW where backhauls may be served before linehauls on any given route. Besides the modelling aspect of this variant we will study its performance implications when compared to the standard VRPBTW using a heuristic algorithm based on Ant Colony Optimization.

Key words: Vehicle Routing Problems, backhauling strategies, Ant Colony Optimization

1 Introduction

A major cost component in many supply chains is associated with the distribution and transportation of goods between member companies of the chain or from the chain to final customers. On average these processes contribute approximately 20% to the total costs of a consumer good. Even if direct costs for transportation are not that significant, distribution is crucial to ensure that the other processes (e.g. production) run smoothly and in a timely manner to satisfy an appropriate service level and responsiveness of a supply chain.

Apart from that some European and national studies report that only about 30% of trucks on European roads are fully loaded, while up to 40% of the traffic volume caused by trucks is due to empty movements.1 In Switzerland, the situation for companies dealing with goods transportation on the road is characterized by the new tax regulation (LSVA), which imposes the

1http://www.vcoe.at/start.asp?pg=publikationen/start.asp?kat0=9
same costs on empty and loaded movements.²

On the one hand, the numbers given above highlight the potential for cost savings through efficient solutions for goods distribution. On the other hand, together these facts imply that improving logistic processes associated with distribution and transportation is both possible and important. Thus, firms have recognized the need to automatize this process and use software to support their distribution process. Part of this software is an optimization tool for routing and scheduling such that research in this area is very interesting from a firms’ point of view.

In fact, this industrial interest is reflected in a huge body of academic research on problems arising in the domain of goods transportation. One of the central problems in this area is the Vehicle Routing Problem (VRP). An overview of the research on the VRP and many of its variants can be found in [1].

Among the most important variants of the VRP are the VRP with Time Windows (VRPTW, for a recent overview of metaheuristic approaches to the VRPTW see [2]), the Multi-Depot VRP (see e.g. [3]) and the VRP with Backhauls and Time Windows (VRPBTW). A research group in Oslo works on a formal representation and algorithms for rich VRPs, which cover a wide range of problems with different constraints and objectives (c.f. [4]). Finally, a recent trend is to come up with robust and flexible algorithms that are capable of solving problems with a variety of real world constraints (see e.g. [5]). A guide to the analysis of such heuristics can be found in [6].

In this paper we address the following problem. Intermediate companies of a supply chain generally deal with suppliers and customers. It has been recognized that significant improvements are possible if a vehicle that delivers goods to customers may also pick up goods from the company’s suppliers. This leads to a mixed problem where some locations need delivery while others require pick up of goods. Another example is the distribution of mineral water from a producer to a retailer (linehauls), which may be coupled with the distribution of empty recyclable bottles from the retailer to the producer (backhauls). These problems are known as Vehicle Routing Problems with Backhauls (VRPB, see e.g. [7]). If the customer visits are constrained to take place within a pre-specified time-window, the problem turns into the VRPBTW mentioned in the previous paragraph.

The main question analyzed in this paper is how to efficiently address this problem. More precisely, three different variants will be considered. First, linehaul and backhaul customers could be separated and treated independently as two different VRPTWs. Second, a combined consideration of linehauls and backhauls with a strict sequence of serving all linehauls before any of the backhauls is possible. This leads to the standard VRPBTW. Third, this sequence can be relaxed to obtain the mixed VRPBTW. All three variants will be discussed in more detail in the next section. While the first two variants have been studied quite extensively, the third strategy is not well understood. Thus, we present a modification of a model developed for

Comparing backhauling strategies in vehicle routing

the VRPB and study its performance implications when compared to the standard VRPBTW.

The remainder of the paper is organized as follows. In the next section we give a more formal description of the problem and review related work. Section 3 summarizes the algorithm used for the analysis carried out in this paper. Computational results and strategy implications for organizing the considered distribution process are given in Section 4 before we formulate some conclusions in Section 5.

2 Vehicle Routing Problems with Backhauls and Time Windows

As stated in the introduction above, we consider the Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW). Given a depot or hub where a fleet of (homogeneous) vehicles is based, a number of customers with known demands have to be served. Customers belong to one of two distinct groups: they either receive goods from the depot, or they send goods to the depot. Each customer must be visited by exactly one vehicle and service has to start during a pre-specified time window. All vehicles’ tours begin and end at the depot within the given time window of the depot. The load of a vehicle must not exceed vehicle capacity at any time along its route. The objective is to find a feasible assignment of customers to vehicle tours that minimizes the required fleet size. Some other possible objectives are to minimize the total distance travelled by all vehicles, or the minimization of the total time spent by all vehicles en route. The VRPBTW is known to be NP-hard in the strong sense. This means that for a given fleet size finding a feasible solution, let alone an optimal one, is NP-complete. This has been proven for the VRPTW (which is a special case of the VRPBTW, where all customers are of one type) in [8].

In principle there are three different strategies to address this problem. Obviously, the simplest one is to split the customers according to their type and to solve the two resulting sub-problems, which are VRPTWs, individually. This is often done in practice, where the delivery of goods is organized by a different department in the company than the pickups. The drawback of this approach is that important synergies may be missed, which generally increases the required fleet size. Still it can be used as a benchmark for the remaining two strategies.

The second approach common in practice is to allow tours that combine pickups and deliveries, however with the restriction that on any tour all deliveries need to be made before any pickup is allowed. The intuition for this approach is that the vehicle is normally loaded in a way that reflects the sequence of delivery customers to ensure efficient unloading. If goods are picked up before the vehicle is completely unloaded additional rearrangements of goods in the vehicle are necessary en route. This increases service times, which may be inefficient and thus to be avoided. This
problem is known as the standard VRPBTW and several approaches to tackle it have been proposed in the academic literature. In [9] a Branch and Bound (B&B) algorithm developed for the VRPTW was extended to cope with backhauling. Further, a set of benchmark problems based on known VRPTW instances was proposed and problems with up to 100 customers were solved to optimality regarding travel times (fleet size is an input for the B&B). Simple construction and improvement algorithms have been proposed in [10], whereas in [11] a Tabu Search algorithm to tackle the problem was proposed. Recently, a Large Neighborhood Search capable of solving a variety of VRPBs was presented in [12]. While the approaches from [10] and [12] considered the objective to minimize fleet sizes first, and to minimize travel times as a second goal, in [11] the minimization of schedule times (which in addition to travel times include service times and waiting times) is considered as the second goal.

While the restriction to perform all deliveries before any pickup eliminates the potential inefficiency of rearranging goods en route, it also reduces the possible synergies of combining pickup and delivery customers, particularly if time windows for pickups are early in the day and deliveries occur late in the day.

To overcome this shortcoming, the third strategy is to relax the restriction concerning the sequence of deliveries and pickups and to allow any schedule of pickup and delivery customer visits that satisfies the capacity constraint. This problem is known as the mixed VRPBTW and has been studied in [12], [13] and [14] from an algorithmic point of view. However, in these papers there is no emphasis on the modelling aspect of the mixed VRPBTW.

In the mixed VRPBTW the following issue arises. If there is no negative influence of picking up goods from a customer before the truck has been completely unloaded then it is obvious that this third strategy can be no worse, and in general will be better than the other two strategies. However, then the justification for the above discussed strategy two is lost. Thus, there needs to be some ‘cost’ associated with mixing pickups and deliveries. This ‘cost’ can be modelled as the time needed to rearrange the remaining delivery load in the truck, once a pickup is made. From a practical point of view, this rearrangement time depends on many different things, e.g. the remaining capacity of the truck, the average load size in the truck and the weight (and/or volume) of the pickup load.

Our model simplifies this situation in the following way. We use two parameters which partially cover the above mentioned factors that influence the rearrangement time. The first one deals with the remaining capacity of the truck. Here we use a threshold parameter $K_{\text{max}}$ that determines the percentage of free capacity required such that a pickup is allowed. For example, if $K_{\text{max}} = 0.5$ then a pickup is allowed if the remaining load of the vehicle is at most 50%. Clearly, if $K_{\text{max}} = 0$, we deal with the standard VRPBTW, whereas as $K_{\text{max}}$ goes to one, the pickup policy becomes more and more liberal. Note, that with respect to this parameter our approach
Comparing backhauling strategies in vehicle routing

is based on and modifies a model originally proposed for the VRPB in [15]. However, the threshold parameter used in [15] was set with respect to the remaining load of the vehicle (rather than with respect to the remaining capacity) and time windows were not considered.

The second parameter we use constitutes a penalty and deals with the increase in the service time for deliveries, once pickups have been made by a vehicle. In this case, the service time for each delivery to be made after the first pickup becomes \( s_i = s_i (1 + \Delta s) \), where \( s_i \) is the original service time for delivery at customer \( i \). Thus, \( \Delta s \) determines the percentage increase in service time due to load rearrangement. If we assume that the original service time somehow reflects the difficulty to unload a customer’s demand, then our approach ensures that the detrimental effect of a pickup is larger if the remaining delivery loads are big, than if they are small. Clearly, as \( \Delta s \) increases, the solution for a given problem instance will more resemble a solution to the standard VRPBTW.

Based on this last observation, a parameter similar to our \( \Delta s \) was considered in [16] in the context of an insertion algorithm for the VRPB. The intuition there was to force backhaul customers towards the ends of the routes, i.e. the parameter was used to ‘shape’ the routes. Further note that no such penalty was considered in [15].

Finally, it is worth mentioning that both parameters in our model have a very different meaning. Whereas \( K_{\text{max}} \) can be chosen by the firm - and this strategic decision reflects the firm’s flexibility - the parameter \( \Delta s \) is an exogenous term that is given by the problem environment. Thus, using these two parameters we can study both the impact of increasing firm flexibility itself and the interplay between a firm’s strategy and its environment.

3 Ant Colony Optimization for Rich VRPs

Since its invention in the early 1990s (see e.g. [17]), Ant Colony Optimization (ACO) has received increasing attention by researchers. Improved versions of the basic algorithm lead to a wide range of successful applications in combinatorial optimization (for an overview see [18]). Recently, a convergence proof for a generalized ACO algorithm has been presented in [19].

The ACO approach is based on the behavior of real ants searching for food. Real ants communicate with each other using an aromatic essence called pheromone, which they leave on the paths they traverse. In the absence of pheromone trails ants more or less perform a random walk. However, as soon as they sense a pheromone trail on a path in their vicinity, they are likely to follow that path, thus reinforcing this trail. More specifically, if ants at some point sense more than one pheromone trail, they will choose one of these trails with a probability related to the strengths of the existing trails. This idea has first been applied to the TSP, where an ant located in a city chooses the next city according to the strength of the artificial trails.
procedure ACO for the VRPBTW{
    Read the input data;
    Initialize parameters and pheromone matrix; (see 3.3)
    repeat { for each ant 
        Construct a solution using the Insertion based Ant System; (see 3.1)
        Improve the solution by Local Search; (see 3.2)
    }
    Update the best found solution (if applicable);
    Update the pheromone matrix; (see 3.3)
} until a pre-specified stopping criterion is met;

Figure 1: The ACO algorithm for the VRPBTW

In this section we will briefly describe our ACO algorithm for the VRPBTW. It was originally proposed in [20] and applied to several variants of the VRP in [21] to show its robustness and flexibility. A high level description of the algorithm is shown in Figure 1.

Before we turn to the presentation of the algorithms’ key components let us briefly highlight the modifications made to the original algorithm presented in [20]. First, the possibility to optimize total route duration rather than total route distance as a second objective (besides minimizing fleet size) was added. Second, the pheromone management was simplified following a scheme presented in [24]. Third, the solution construction was modified to be able to take into account the possibility to serve backhauls prior to linehauls.

3.1 Insertion based Ant System

As mentioned above, we use the algorithm proposed in [20]. The randomized solution construction at the core of this ACO algorithm is derived from the I1 insertion heuristic originally presented in [22] for the VRPTW.

In this algorithm routes are constructed sequentially one by one. First, the unrouted customer farthest from the depot is selected as a seed customer to initialize the current route. Next, other unrouted customers are sequentially inserted into this route according to a cost criterion that takes into account both, the difficulty to insert a customer as well as the detour caused by inserting the customer. Once no more insertions are feasible with respect to time window, capacity or tour length constraints, another route is initialized with a seed customer and the insertion procedure is repeated with the remaining unrouted customers. The algorithm stops when all customers are assigned to routes.

In order to use the algorithm described above within the framework of our ACO algorithm an adaptation of the decision making is necessary. More precisely, some of the deterministic decisions have to be replaced by probabilistic choices.
First, to initialize a tour, seed customers are not chosen deterministically but probabilistically according to their distance from the depot.

Second, inserting further customers on the current tour is done using a roulette wheel selection over all unrouted customers. The decision rule used can be written as

\[ P_i = \frac{\max_{j \in R_i} [\eta_{ij} \cdot \frac{T_{ij} + T_{jk}}{2 \tau_{ij}}]}{\sum_{h \in N_u} \max_{j' \in R_h} [\eta_{j'i} \cdot \frac{T_{ij'} + T_{ik'}}{2 \tau_{ij'}}]} \quad \forall i \in N_u \tag{1} \]

where \( P_i \) is the probability that customer \( i \) is chosen to be inserted in the current route. Further, \( \eta_{ij} \) denotes the attractiveness of inserting customer \( i \) immediately after customer \( j \) and \( \tau_{ij} \) denotes the pheromone concentration on the arc connecting locations (customers or depot) \( j \) and \( i \). The pheromone concentration \( \tau_{ij} \) contains information about how good visiting two customers \( i \) and \( j \) immediately after each other was in previous iterations. Further, \( k \) (\( k' \)) is the customer visited immediately after customer \( j \) (\( j' \)) in the current solution, \( R_i \) denotes the set of customers assigned to the current tour after which customer \( i \) could feasibly be inserted and \( N_u \) denotes the set of unrouted customers. Note, that constructing the set \( R_i \) actually deals with potential violations of time windows or the capacity constraint. For example, consider the case of the standard VRPBTW. Let \( i \) be a linehaul customer, then \( R_i \) comprises only the set of linehaul customers assigned to the current tour as \( i \) can not be inserted after any backhaul customer. Similarly, time window or capacity constraints are dealt with.

The attractiveness of inserting customer \( i \) immediately after customer \( j \) is computed using an extension of the function proposed in \[22\] for the VRPTW. Formally, for a feasible position \( j \) the attractiveness of inserting \( i \) is given by

\[ \eta_{ij} = \max \{0, d_{0i} - (d_{ji} + d_{ik} - d_{jk}) + \gamma \cdot type_i\} \quad \forall i \in N_u, \forall j \in R_i, \]

where \( d_{0i} \) denotes the distance between the depot and customer \( i \) and \( type_i \) is a binary indicator variable denoting whether customer \( i \) is a linehaul (\( type_i = 0 \)) or a backhaul customer (\( type_i = 1 \)). This allows us to discriminate between linehaul and backhaul customers and this is done using parameter \( \gamma > 0 \). Apart from that, the attractiveness \( \eta_{ij} \) can become zero if the detour associated with inserting customer \( j \) is greater than its distance from the depot. This may lead to situations where all feasible insertions on a vehicle have zero attractiveness and are thus abandoned. Clearly, this is counterproductive in terms of the required fleet size and \( \gamma > 0 \) alleviates this problem.

Following this approach, the probabilistically chosen customer \( i \) is deterministically inserted at its best position along the current route.

---

3Note, that the same pheromone utilization is done for route initialization, thus augmenting the attractiveness of initializing a route with an unrouted customer \( i \) by the search-historic information.
3.2 Local Search

After an ant has constructed its solution, we apply local search to improve the solution quality. In particular, we first apply a Swap neighborhood operator, where two customers exchange their positions until no more improvements are possible. Then we use the Move neighborhood operator where one customer is relocated in its own route or in another route. Again, this operator is applied until no more solution improvements are possible. For both operators we use a first improvement strategy and we do not accept infeasible solutions. General versions of both operators, where \( \lambda \) adjacent customers can be moved or swapped, have been proposed in [23].

3.3 Pheromone initialization and update

In the constructive phase of the ACO algorithm decisions are based on both heuristic information and the pheromone values as described above. At the end of each iteration, i.e. once all ants have gone through solution construction and local search, the pheromone update procedure is applied to these pheromone values. The update used for the algorithm presented in this paper is a variant of the approach proposed in [24]. It can be written as

\[
\tau_{ij} := \rho \tau_{ij} + (1 - \rho) \Delta \tau_{ij}^* \quad \forall (ij) \in E
\]

where \( E \) is the set of all edges, \( 0 \leq \rho \leq 1 \) is called the trail persistence and \( \Delta \tau_{ij}^* \) is the amount of reinforcement, which is defined as

\[
\Delta \tau_{ij}^* := \begin{cases} 
1 & \text{if } (ij) \in S^* \\
0 & \text{otherwise}
\end{cases}
\]

where \( S^* \) is the best solution found up to the current iteration (regardless if it was found in the current iteration or earlier). Note that pheromone values are unit-free and thus independent of monotonous transformations of the objective function value. Together with the fact that at the beginning of the run, the pheromone values are initialized to 1, i.e.

\[
\tau_{ij} = 1 \quad \forall (ij) \in E
\]

this update strategy implies that the pheromone values now have a well defined domain, namely \( \tau_{ij} \in [0, 1] \). Note that a more general version of this update is the so called Hypercube Framework proposed in [25].

4 Numerical Analysis

In this section we turn to the numerical analysis of the performance of the different strategies for backhauling, as well as to the influence of the parameters \( K_{max} \) and \( \Delta s \) in case of the mixed VRPBTW.
The ACO algorithm used for this numerical study has been implemented in C. Its parameters, as already proposed in [20], are \( \frac{n}{2} \) ants (where \( n \) is the number of customers), \( \rho = 0.95 \) and \( \gamma = 13 \). The stopping criterion was 2.5 minutes of CPU time on a Pentium 4 with 1.5GHz. Each instance was solved 5 times and all results presented below are based on averages over these 5 runs.

As mentioned above, the objective for time constrained routing problems is generally to first minimize the fleet size required to serve all customers and given a minimal fleet size to minimize the total distance travelled. This lexicographic ordering of objectives was established by minimization of the following objective function:

\[
L = M \cdot FS + TT,
\]

(5)

where \( L \) denotes the total costs of a solution, \( FS \) denotes the fleet size found, and \( TT \) corresponds to the total travel time (or distance). Sometimes, e.g. in [11] the secondary objective was to minimize the total route duration of all tours. This can be established by exchanging \( TT \) with \( TD \), the total route duration in equation (5). The parameter \( M \) has to be chosen in a way to ensure that a solution that saves a vehicle always outperforms a solution with a higher fleet size. More precisely, in our experiments setting \( M = 10000 \) was sufficient to achieve this goal.

Let us next provide some information about the problem instances, before we analyze the performance of the different backhauling strategies.

### 4.1 The benchmark problem instances

The benchmark problem instances we used for our numerical tests were proposed in [9]. They are based on the first five instances from the r1 set, namely r101 to r105, originally developed for the VRPTW and first presented in [22].

Each of these problems consists of 100 customers to be served from a central depot. The customers are located randomly around the depot. Service has to take place within a short time horizon (230 time units), and vehicle capacities are fairly loose when compared with the time window requirements at the customers. These time window requirements are varying in the data sets and the average time window length ranges from 10 up to 148.3 time units.

Given these data sets, 10%, 30% and 50% of the customers were randomly chosen to be backhaul customers with unchanged quantities, thus creating 15 different 100 customer instances. According to the number of backhaul customers, an \( a \) (for 10% backhauls), a \( b \) (for 30% backhauls) or \( c \) (for 50% backhauls) was added to the original problem name. Thus, the instance \( r102b \), corresponds to the original VRPTW instance \( r102 \), where 30% of the customers have been changed to backhaul customers.
4.2 Analyzing the performance of different backhauling strategies

In this section we compare and analyze the performance of the three backhauling strategies discussed in Section 2. Whereas the first two strategies are clearly defined, the third strategy requires the choice of the two parameters $K_{\text{max}}$ and $\Delta s$.

The standard setting we use is to set $K_{\text{max}} = 1$ and $\Delta s = 0.5$. This corresponds to a situation, where the mix of linehauls and backhauls on a route is completely free as long as all constraints are satisfied. In particular, these are the capacity constraints which have to be monitored all along the route, and the time constraints which may be violated due to the increasing service times, induced by $\Delta s = 0.5$, for linehauls which are served after backhauls.

Let us first compare the solution quality obtained for the three strategies. In Table 1 the three strategies are denoted as

- **VRPTW ($\Delta s, K_{\text{max}}$ not applicable)** - this is the first strategy, where linehauls and backhauls are considered as two separate VRPTWs
- **VRPBTW ($\Delta s$ not applicable, $K_{\text{max}} = 1$)** - this is the second strategy, where on each route all linehauls must precede all backhauls
- **mixed VRPBTW ($\Delta s = 0.5, K_{\text{max}}$ varying)** - this is the third strategy, where the sequence of linehauls and backhauls is free

and the results for each of the 15 problem instances are shown. The last row of Table 1 provides cumulative numbers over all instances.

From Table 1, we observe that the difference between the VRPTW strategy and the VRPBTW strategy is very small. Particularly for the fleet size this difference is approximately 1%, while for the total travel times it is somewhat larger at 3%. The reason for this lies in the structure of the problem instances. In the original problem instances, customer time windows are regularly distributed over the course of the planning horizon. In the modification of the instances for the VRPBTW customers were randomly chosen to be backhaul customers such that in the resulting instances both linehaul and backhaul customers feature time windows which are evenly distributed in time. Thus, for both linehauls and backhauls reasonable tours can be built separately and the improvements of joint consideration are expected to be small.

Given this observation, a strong effect of relaxing the linehaul-backhaul precedence should be expected. Indeed, from Table 1 we see that the results for the mixed VRPBTW are much better than those for the competing two strategies. Improvements are around 12% in both fleet size and total travel times.

These results suggest that there is a strong relationship between the time windows and the load type, i.e. linehaul or backhaul, and that this relationship will influence of the relative performance of the three strategies.
### Table 1: Comparison of different backhauling strategies

<table>
<thead>
<tr>
<th>VPRBTW</th>
<th>FS</th>
<th>TT</th>
<th>VPRBTW</th>
<th>FS</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inst.</td>
<td></td>
<td></td>
<td>Inst.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r101a</td>
<td>22.0</td>
<td>1918.6</td>
<td>r101b</td>
<td>23.4</td>
<td>2043.4</td>
</tr>
<tr>
<td>r101c</td>
<td>24.0</td>
<td>2098.6</td>
<td>r101d</td>
<td>21.0</td>
<td>1733.8</td>
</tr>
<tr>
<td>r101e</td>
<td>24.0</td>
<td>1998.6</td>
<td>r102a</td>
<td>22.0</td>
<td>1852.0</td>
</tr>
<tr>
<td>r102b</td>
<td>22.0</td>
<td>1833.2</td>
<td>r102c</td>
<td>22.0</td>
<td>1766.8</td>
</tr>
<tr>
<td>r102d</td>
<td>22.0</td>
<td>1633.2</td>
<td>r102e</td>
<td>22.0</td>
<td>1526.2</td>
</tr>
<tr>
<td>r103a</td>
<td>17.0</td>
<td>1425.6</td>
<td>r103b</td>
<td>16.0</td>
<td>1480.6</td>
</tr>
<tr>
<td>r103c</td>
<td>17.0</td>
<td>1285.6</td>
<td>r103d</td>
<td>16.0</td>
<td>1235.6</td>
</tr>
<tr>
<td>r104a</td>
<td>12.0</td>
<td>1256.2</td>
<td>r104b</td>
<td>12.0</td>
<td>1285.6</td>
</tr>
<tr>
<td>r104c</td>
<td>12.0</td>
<td>1236.2</td>
<td>r104d</td>
<td>12.0</td>
<td>1283.2</td>
</tr>
<tr>
<td>r105a</td>
<td>16.0</td>
<td>1666.4</td>
<td>r105b</td>
<td>16.0</td>
<td>1666.4</td>
</tr>
<tr>
<td>Sum</td>
<td>208.8</td>
<td>24569.2</td>
<td>FS, fleet size</td>
<td>226.0</td>
<td>23692.2</td>
</tr>
<tr>
<td>TT, travel times</td>
<td>206.0</td>
<td>23992.2</td>
<td>FS</td>
<td>238.4</td>
<td>24232.2</td>
</tr>
<tr>
<td></td>
<td>TT</td>
<td>208.8</td>
<td></td>
<td>TT</td>
<td>206.0</td>
</tr>
<tr>
<td></td>
<td>24569.2</td>
<td></td>
<td></td>
<td>23992.2</td>
<td></td>
</tr>
</tbody>
</table>
More specifically, if the linehauls were to be picked up over the course of the morning and the backhauls were to be picked up in the afternoon, the relative difference between the standard VRPBTW and the mixed VRPBTW strategy would be much smaller, while the VRPTW strategy would perform much worse. In practice, time windows are often subject of negotiation between the customers and the supplier. Knowledge about the influence of certain time window decisions on the performance of the routing strategy should help the sales people to be able and offer time windows to customers that are more favorable for the operative planning. On the other hand, such an offer could go ahead with a possible price reduction for the customer leading to a win-win situation. The investigation of this relationship is left for future research.

Let us now turn to the influence of the two parameters of the mixed VRPBTW strategy. First, we will analyze the performance implications of different levels of $K_{max}$. More precisely, Table 1 shows results for $K_{max} = 1$, $K_{max} = 0.75$, $K_{max} = 0.5$ and $K_{max} = 0.25$, thus for reduced levels of flexibility. Note, that $K_{max} = 0$ is the case where the remaining linehaul load of the vehicle has to be zero, which corresponds to the standard VRPBTW. For this analysis, the second parameter $\Delta s$ was left unaltered at $\Delta s = 0.5$.

From Table 1 we observe that the parameter $K_{max}$ has comparatively little influence on the performance of the mixed VRPBTW strategy. In fact, even for the very restrictive policy $K_{max} = 0.25$ the performance degradation is only around 1% when compared with the most liberal case $K_{max} = 1$. The reason for this effect can again be found by looking at the problem instances. As stated above, these instances feature a short planning horizon and rather tight time windows when compared with vehicle capacity. Thus, vehicles are fairly lightly loaded in general and the actual value of threshold $K_{max}$ plays a minor role. However, together with the fact that using a mixed strategy outperforms the standard VRPBTW by 12% this shows that already a small increase in scheduling flexibility pays off significantly.

Finally, let us look at the effect of the rearrangement costs on the performance of the mixed VRPBTW strategy. To that end we compare three values of $\Delta s$, namely $\Delta s = 0.5$, which was our standard setting, with $\Delta s = 0.25$, i.e. a case where rearranging loads is less costly and $\Delta s = 1$, a case where the costs of mixing backhauls and linehauls is larger. For this analysis we have set the load threshold to $K_{max} = 0.25$ and the average results are shown in Figure 2.

Clearly, the gains from the scheduling flexibility erode as the rearrangement costs increase. A backhaul customer inserted before a set of linehaul customers leads to an increase in the service times at these linehaul customers. If this effect is increased, i.e. if rearrangement costs are larger, the probability that the resulting solution becomes infeasible with respect to time windows also increases. Thus, with increasing rearrangement costs it becomes less and less favorable to serve backhaul customers before linehaul
Figure 2: Influence of rearrangement costs on the relative performance of the mixed \textit{VRPBTW} strategy

customers. Still, even for the large penalty of a twofold increase in service times the improvements are between 5 and 7 \% in our experimental results and include both reductions in fleet size and total travel times.

Obviously, these results provide only a starting point for further analysis of this relationship between scheduling flexibility and the problem characteristics, most prominently the rearrangement costs. However, our results show the potential merit of such an analysis for improving the efficiency of VRPBTW like distribution processes.

4.3 Comparison of our ACO with existing results for the standard VRPBTW

Let us now turn to the analysis of the solutions found with respect to absolute solution quality. As in the past the majority of work on the VRPBTW has been on the standard version we will compare our ACO with several other approaches on the basis of this problem.

Most approaches proposed in the literature have been tested on the objective to minimize fleet size first, and to minimize total travel times as a secondary objective. Table 2 shows a comparison of our ACO with three competing approaches. These are the constructive algorithm and the local searches (denoted by \textit{LS}) presented in [10], the Guided Local Search (\textit{GLS}) proposed in [14] and the Large Neighborhood Search (\textit{LNS}) from [12].

More precisely, in Table 2 we show for each of the 15 instances and for each of the competing algorithms the required fleet size and the total travel time of the best reported solution. Further, for our ACO, we show both
the average results obtained over 5 runs as well as the best results obtained throughout our analysis of the backhauling strategies. The last row of the table reports the aggregate numbers for the approaches.

<table>
<thead>
<tr>
<th>Instance</th>
<th>LS best</th>
<th>GLS best</th>
<th>LNS best</th>
<th>Avg.</th>
<th>ACO best</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>best</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FS TT</td>
<td>FS TT</td>
<td>FS TT</td>
<td>FS TT</td>
<td>FS TT</td>
</tr>
<tr>
<td>rl01a</td>
<td>24</td>
<td>1842.3</td>
<td>24</td>
<td>1848.04</td>
<td>22</td>
</tr>
<tr>
<td>rl01b</td>
<td>24</td>
<td>1928.6</td>
<td>24</td>
<td>2034.81</td>
<td>23</td>
</tr>
<tr>
<td>rl01c</td>
<td>25</td>
<td>1937.6</td>
<td>25</td>
<td>2057.05</td>
<td>24</td>
</tr>
<tr>
<td>rl02a</td>
<td>20</td>
<td>1854.1</td>
<td>19</td>
<td>1913.19</td>
<td>19.4</td>
</tr>
<tr>
<td>rl02b</td>
<td>21</td>
<td>1764.3</td>
<td>22</td>
<td>1750.70</td>
<td>22</td>
</tr>
<tr>
<td>rl02c</td>
<td>21</td>
<td>1745.3</td>
<td>22</td>
<td>1775.76</td>
<td>23</td>
</tr>
<tr>
<td>rl03a</td>
<td>15</td>
<td>1371.6</td>
<td>15</td>
<td>1387.57</td>
<td>15.8</td>
</tr>
<tr>
<td>rl03b</td>
<td>16</td>
<td>1477.6</td>
<td>15</td>
<td>1390.33</td>
<td>15</td>
</tr>
<tr>
<td>rl03c</td>
<td>17</td>
<td>1543.3</td>
<td>17</td>
<td>1458.48</td>
<td>17</td>
</tr>
<tr>
<td>rl04a</td>
<td>13</td>
<td>1220.3</td>
<td>11</td>
<td>1084.17</td>
<td>11.4</td>
</tr>
<tr>
<td>rl04b</td>
<td>12</td>
<td>1303.5</td>
<td>11</td>
<td>1154.84</td>
<td>17</td>
</tr>
<tr>
<td>rl04c</td>
<td>13</td>
<td>1346.6</td>
<td>11</td>
<td>1191.28</td>
<td>13</td>
</tr>
<tr>
<td>rl05a</td>
<td>17</td>
<td>1553.4</td>
<td>17</td>
<td>1690.54</td>
<td>16.2</td>
</tr>
<tr>
<td>rl05b</td>
<td>18</td>
<td>1708.7</td>
<td>17</td>
<td>1667.92</td>
<td>16</td>
</tr>
<tr>
<td>rl05c</td>
<td>18</td>
<td>1657.4</td>
<td>17</td>
<td>1710.18</td>
<td>17.3</td>
</tr>
<tr>
<td>Sum</td>
<td>274</td>
<td>24052.9</td>
<td>259</td>
<td>23418.71</td>
<td>268</td>
</tr>
<tr>
<td>Avg. Time</td>
<td>17 sec.</td>
<td>39 sec.</td>
<td>114 sec.</td>
<td>75 sec.</td>
<td></td>
</tr>
</tbody>
</table>

FS...fleet size
TT...travel times

LS was run on a NeXT Machine with a 33MHz Motorola 68040 processor
GLS was run on a 450MHz PC
LNS was run on a PC with 1.5GHz Pentium 4 processor
ACO was run on a PC with 1.5GHz Pentium 4 processor

Table 2: Comparison of different approaches for minimizing total travel times as second objective

Let us clarify a number of details about the results shown in Table 2. First, the results of the LS were obtained using 5 different versions of local search algorithms. On the other hand, computation times for this approach are by far the smallest. Thus, the merit of LS clearly lies in proposing useful local search operators and in giving reasonable solutions very quickly.

Second, for the GLS we do not know results for all instances. While this algorithm does not perform particularly well on these problem instances - it is in general even outperformed by the simple LS - its merit is that it has been applied to problems of different types and its flexibility partly offsets its poor performance in our comparison.

Third, the LNS results were obtained from 10 runs for each instance. The average computation time was reported to be around 110 seconds on a 1.5GHz Pentium 4 running under Linux. Thus, these computation times are in the same order as the computation times of our ACO. Given the fact that the LNS was designed for and successfully applied to a wide range of VRPBTWs the results are impressive. While we do not have complete information about the variability of the approach, we know from [12] that the average cumulated fleet size is 259.9 vehicles. Thus, the approach also seems to be very robust.

Finally, let us look at our ACO. Clearly, the performance of the ACO lies between the excellent behavior of the LNS and the rather poor results obtained by the LS and GLS. In fact, the best ACO results obtained show that our ACO comes close to the results of the LNS, matching the fleet size...
Comparing backhauling strategies in vehicle routing

in 13 cases and showing an average deviation in travel times of less than 2%.

Finally, as already discussed in Section 2 a possible second objective, besides minimizing fleet size as a first objective, is to minimize total route duration rather than total distance. Total route duration includes travel times, service times and waiting times. This objective was tackled in [11] with a Tabu Search (TS) approach.

Table 3 shows how our ACO performs when compared with the Tabu Search. This comparison is based solely on the three VRPBTW instances derived from r103, as this is the only instance for which results are reported in [11].

<table>
<thead>
<tr>
<th>Instance</th>
<th>TS avg. 10 runs</th>
<th>ACO avg. 5 runs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FS TD</td>
<td>FS TD</td>
</tr>
<tr>
<td>r103a</td>
<td>17 2765,1</td>
<td>16 2494,80</td>
</tr>
<tr>
<td>r103b</td>
<td>16 2784,5</td>
<td>15,2 2607,00</td>
</tr>
<tr>
<td>r103c</td>
<td>17,9 3008,6</td>
<td>17 2667,00</td>
</tr>
</tbody>
</table>

FS...fleet size
TD...total route duration

Table 3: Comparison of different approaches for minimizing total route duration as second objective

Table 3 clearly shows the superiority of our ACO over the Tabu Search with respect to both objectives. On the other hand, the Tabu Search was run for much shorter computation times. In [11] computation times of approximately 70 seconds on a Sun Sparc 10 workstation are reported. Thus, it is again hard to provide a fair comparison of results. Nevertheless the quality of our ACO results seems to be very good.

Finally, let us conclude this section with some more considerations about our ACO. We have shown, that the ACO is competitive in terms of solution quality with state-of-the-art approaches. Clearly, as already mentioned in the introduction, this is not the only measure to evaluate an algorithm. Based on the scheme proposed in [6], other important issues are flexibility and simplicity. Flexibility means that the algorithm is applicable to a range of problems and problem modifications can be dealt with easily. In this paper our algorithm was shown to be able to deal with different constraints concerning the linehaul-backhaul precedence and different objectives. Further, in [21] the algorithm was tested on a range of problems including the VRP, the VRPTW and the VRPB and results were shown to be near best known results for each of these classes. Thus, we believe that one of the advantages of our ACO is its flexibility, which comes at little cost in terms of robustness concerning solution quality. Simplicity is generally seen as the
ease to understand the working principle of an algorithm, the algorithms complexity in terms of parameters and the skill needed to implement it. Our ACO is based on a very simple learning paradigm, and the constructive and local search heuristics are well known and easy to comprehend. In fact, we use only basic operators. Moreover, our algorithm features, apart from the permissible computation time, just three parameters, which are also easy to understand, namely the population size, the rate of forgetting past experience and the discrimination factor between linehauls and backhauls. Summarizing we are confident that our algorithm performs quite well with respect to these issues and justifies further investigation on vehicle routing problems.

5 Conclusions and Future Research

In this paper, we have dealt with an operational problem arising in the planning of distribution and transportation within a supply chain. More precisely, the efficient routing of vehicles faced with two types of demands, namely delivery orders from the depot to customers and pickup orders from customers to the depot, was analyzed. This problem is known as the Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW) and different strategies to address it exist. Using a reimplementation of an existing Ant Colony Optimization (ACO) algorithm we compared three different strategies with respect to backhauling.

The results obtained show that a strategy, where linehauls and backhauls may be mixed to some extent along a route significantly outperforms both a strategy, where linehauls and backhauls are considered separately and a strategy, where linehauls and backhauls may be served by the same vehicle subject to the strict sequence of all linehauls preceding all backhauls. Additionally, possible improvements were observed to exist for a quite broad range of penalties for rearranging goods en route. Moreover, it is shown that at least for the class of problem instances tackled only a moderate relaxation of the sequence requirement between linehauls and backhauls is necessary. In fact, most of the cost reduction (in terms of fleet size and travel time) was achieved by allowing pickups once the remaining load of the vehicle is less than 25%.

Clearly, more work is needed to fully understand the implications of mixing linehauls and backhauls along a route. First, different approaches to model the rearrangement of goods and the associated costs should be considered. Second, as pointed out in Section 4.2 the strategic interplay between the time windows and the effects of different backhauling strategies should be analyzed more deeply in future work. Third, the actual backhaul demand of a customer is often known only upon arrival at the customer. Thus, studying the stochastic version of the problem is also interesting both from a scientific and a real world point of view.

Finally, it should be interesting to study the VRPBTW as a true multi-
objectives problem to further enhance the applicability of the approach to real world situations. An existing approach by Jozefowiez et al. [26] studies this possibility for the basic VRP. Their objectives are the total tour length and a load balancing criterion that asks for tours with little difference in tour length. Other possibilities are to modify the approach described in [27] for the VRPTW, or to use the multi-colony Ant System proposed in [28].

Acknowledgments

The authors wish to thank Sabine Emch for re-implementing the algorithm and performing the numerical tests. Further, we are grateful to the 3 anonymous referees for their helpful comments.

References


Comparing backhauling strategies in vehicle routing


