A Heuristic Approach for the Swap-Body Vehicle Routing Problem

Oliver Lum, Ping Chen, Xingyin Wang, Bruce Golden, Edward Wasil
Department of Applied Mathematics and Scientific Computation, University of Maryland, College Park oliver@math.umd.edu
Department of Logistics Management, Nankai University, Tianjin, China chenpingteda@nankai.edu.cn
Department of Applied Mathematics and Scientific Computation, University of Maryland, College Park wangxy@umd.edu
Robert H. Smith School of Business, University of Maryland, College Park bgolden@rhsmith.umd.edu
Kogod School of Business, American University ewasil@american.edu

Abstract The Swap-Body Vehicle Routing Problem is a variant of the well-studied vehicle routing problem which incorporates additional complicating factors encountered in practice. In this problem, a homogeneous fleet of an unlimited number of trucks, semi-trailers, and swap-bodies are used to serve a set of customers. Some customers may only be served by trucks due to a set of location-dependent accessibility constraints. Four types of operations (park, pickup, swap, and exchange) can be performed at designated swap locations, each with an associated time cost. The objective consists of fixed and variable costs for vehicle components involving service and swap action times. In this paper, we present a survey of related problems, and propose a local search based heuristic approach for this challenging problem. Computational results earned a place in the finals of a competition against nearly 30 teams of international researchers.

Keywords Vehicle Routing Problem, Heuristics

1. Introduction
The Swap-Body Vehicle Routing Problem (SB-VRP) is a variant of the vehicle routing problem that was proposed by the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog) and the PTV Group, a German company. Teams were invited to solve the SB-VRP using operations research methods in the inaugural VeRoLog Solver Challenge (VSC 2014) that is described in [15].

In the SB-VRP, each customer requires delivery of a specified quantity of an item at one time (i.e., there are no split deliveries), and has a service time that is correlated with the delivery quantity. In addition, accessibility constraints create three categories of customers: some customers cannot be served by a train and can only be served by a truck (reflecting space or weight constraints in practice), and are, therefore, referred to as ‘truck-only’ customers; others with demand greater than a single swap-body’s capacity must be serviced by a train, and are referred to as ‘train-only’ customers; the remaining customers can be served by a truck or a train, and are labelled ‘train-possible’.

These customers are serviced by a vehicle fleet that is homogeneous, departs from a single depot, and has three components available in unlimited quantities: a truck, a semi-trailer, and a swap-body (see Figure 1). With these components, there are two valid vehicle
configurations: a truck which carries a single swap-body on its flatbed; and a train which consists of a truck, a semi-trailer, and two swap-bodies. In this context, a swap-body is simply a container with fixed capacity that holds the items to be delivered to the customers.

En route to a customer, a train can detach a semi-trailer with a swap-body, and leave it at a specified location to be reattached at a later time. In this way, a train that departed the depot with two swap-bodies can still serve a ‘truck-only’ customer. It is important to note that swap-bodies cannot ever pick up a swap-body that belonged to another train (i.e., a vehicle must return to the depot carrying the same swap-bodies as it departed with).

There is a fixed cost associated with the use of a truck, semi-trailer, and swap-body. Additionally, for each vehicle, there is a cost per unit time associated with driver salary, and a cost per unit distance (that depends on the vehicle type) that reflects a transportation resource cost (e.g., gas cost). There is also a maximum duration for a route. The goal is to develop a set of routes that minimizes the cost function. The SB-VRP for the challenge is described in [16].

The paper is organized as follows. In Section 2, we review the literature on the SB-VRP. In Section 3, we formally define the SB-VRP. In Section 4, we develop a heuristic for the SB-VRP. We report our computational results in Section 5. Conclusions are given in Section 6.

2. Literature Review

Although the SB-VRP is a new problem, related problems have been studied in the literature. One common element in all of these problems is the composition of the fleet with trucks (vehicles or lorries) and trailers (tanks or semi-trailers). Some routes are traversed by trucks only, while others are traversed by a truck coupled to a trailer. Trailers may need to be parked at locations before serving some truck-only customers due to limited maneuvering space or limited access over narrow roads. There are many differences in problem characteristics, constraints, and objectives. In Table 1, we summarize problems related to the SB-VRP from 1995 to 2014. The relationships among the problems are shown in Figure 2.
Table 1: Problems related to the SB-VRP in the literature from 1995 to 2014

<table>
<thead>
<tr>
<th>Problem</th>
<th>Source</th>
<th>Characteristics</th>
<th>Constraints</th>
<th>Objective Function</th>
<th>Solution Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial Accessibility Constrained Vehicle Routing Problem (PACVRP)</td>
<td>Semet [14]</td>
<td>Number of trucks &gt; number of trailers, Number of trailers to use is determined Only one subtour can depart each parking place</td>
<td>Vehicle capacity Customer accessibility</td>
<td>Travel cost</td>
<td>Cluster-first, route-second heuristic</td>
</tr>
<tr>
<td>VRP with Trailers (VRPT)</td>
<td>Gerdessen [7]</td>
<td>Customers all with demand 1 Trailer is parked precisely once</td>
<td>Vehicle capacity Vehicle maneuvering time</td>
<td>Total travel time + total maneuvering time</td>
<td>Three constructive heuristics + local search</td>
</tr>
<tr>
<td>Problem</td>
<td>Source</td>
<td>Characteristics</td>
<td>Constraints</td>
<td>Objective Function</td>
<td>Solution Method</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Truck and Trailer Routing Problem with Time Windows (TTRPTW)</td>
<td>Lin et al. [12]</td>
<td>Truck-only customers Number of trucks ≥ number of trailers</td>
<td>Vehicle capacity Customer accessibility Time windows for delivery</td>
<td>Total distance</td>
<td>Simulated annealing</td>
</tr>
<tr>
<td></td>
<td>Derigs et al. [3]</td>
<td>TTRP with and without load transfer between truck and trailer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relaxed Truck and Trailer Routing Problem (RTTRP)</td>
<td>Lin et al. [11]</td>
<td>Unlimited number of trucks and trailers</td>
<td>Vehicle capacity Customer accessibility</td>
<td>Fixed cost variable cost</td>
<td>Simulated annealing</td>
</tr>
<tr>
<td>Vehicle Routing Problem with Trailers and Transshipments (VRPTT)</td>
<td>Drexl [6]</td>
<td>Transshipment locations, No fixed assignment of a trailer to a truck (lorry) Transshipment time depends on the amount of load that is transferred</td>
<td>Vehicle capacity Customer accessibility Time windows for delivery</td>
<td>Fixed cost + variable cost</td>
<td>Branch-and-cut</td>
</tr>
</tbody>
</table>
The partial accessibility constrained vehicle routing problem (PACVRP) was proposed by Semet [14] in 1995. In this problem, the number of components (trucks, trailers, etc.) available is limited, and all available trucks are used, while the number of trailers used needs to be determined. Furthermore, at most one subtour is allowed to depart from each parking place for trailers. Semet presented an integer programming formulation, and built a cluster-first, route-second heuristic for the PACVRP. The clustering phase of the heuristic was solved using a branch-and-bound algorithm with Lagrangian relaxation for the trailer assignment.

The vehicle routing problem with trailers (VRPT) was introduced by Gerdessen [7]. In the VRPT, two important assumptions are made for simplification. Every customer has unit demand and each trailer can only be parked exactly once. Customers are not designated as truck-only or train types. Instead, customers are assigned different maneuvering times. These times are associated with the marginal cost of serving a customer with a trailer as opposed to without (e.g., if it takes 3 units of time to serve a customer with a trailer attached, but only 1 unit without, then the maneuvering time for this customer would be 2). In this way, customers that are difficult to reach are modeled with larger maneuvering times. There are no special sites for parking or pick-up, and each customer site can be used as a parking place. Gerdessen developed three constructive heuristics and an improvement heuristic method based on local search for the VRPT.

The truck and trailer routing problem (TTRP) was introduced by Chao [2], and it is the most-studied problem (from Table 1) in the literature [1, 2, 10, 13, 17, 19, 18]. Chao [2] described the TTRP as a real-world variant of the vehicle routing problem in which a fleet of $m_k$ trucks and $m_l$ trailers ($m_k \geq m_l$) serves a set of customers. There are two types of customers. A vehicle customer can be served either by a vehicle (a truck pulling a trailer) or by a truck, while a truck customer can only be served by a truck. The capacity of a truck and a trailer are denoted by $Q_k$ and $Q_l$, respectively. Each customer location can be used as a parking place, as in the VRPT. A solution to the TTRP is shown in Figure 3. We see that there are three kinds of routes: a pure truck route (PTR), a pure vehicle route (PVR), and a complete vehicle route (CVR) [13]. A route must satisfy the capacity constraints, but is not subject to any time constraints as in the SB-VRP. The objective is to minimize the total distance traveled by the fleet over all three types of routes.

Since the TTRP is a generalization of the VRP, it is NP-hard. Research on the TTRP has focused on constructive heuristics [2, 13], tabu search [2, 13], simulated annealing [10], ant
Colony systems [19] and hybrid heuristics [17, 18, 1]. Chao [2] proposed a solution construction method and an improvement method that combined tabu search and the deviation concept found in deterministic annealing. Chao developed 21 test problems that were converted from classic VRP benchmark instances, and then used to evaluate algorithm performance. Scherer [13] presented two new construction heuristics (T-cluster and T-sweep) and a heuristic based on tabu search. The initial solutions generated by the construction methods were improved by the tabu search heuristic. Lin et al. [10] developed an effective simulated annealing heuristic for solving the TTRP that was competitive with the tabu search procedure in [2, 13], both in terms of solution quality and computational time. Carania and Guerriero [1] proposed a hybrid heuristic that combined mathematical programming methods with local search procedures. The framework split the problem into two subproblems, which assigned customers to routes and then defined the routes given the customers. Local search dealt with the infeasibility of the solution by the route assignment model. Diverse solutions were obtained via a multi-start metaheuristic that operated over the iterative mathematical programming formulations and local search procedures. Villegas et al. [17] presented a hybrid metaheuristic method, based on a greedy randomized adaptive search procedure (GRASP), variable neighborhood search (VNS), and path relinking (PR) that employed a route-first, cluster-second solution procedure. Recently, they developed a new matheuristic [18] that combined an exact method with a heuristic method. In this approach, local optima were generated by a hybrid method using GRASP and iterated local search (ILS). The solution is generated by using a two-stage metaheuristic to solve a mathematical programming formulation of the TTRP, using routes of the local optima (from the hybrid method) as columns to solve a set partitioning problem. Both approaches performed quite well in terms of speed and solution quality. Villegas et al. [17] reported an average deviation of .36% from the best-known solutions, with an average run time of 14.58 minutes. Villegas et al. [18] reported comparable solution quality (between .30% to .50% depending on the benchmark instances), in about a third of the time (5.03 minutes). Yu et al. [19] proposed a two-stage method for the TTRP that combined an ant colony system with an improvement procedure based on local search.

Several extensions of the TTRP have been studied recently. Lin et al. [11] proposed a relaxed TTRP (denoted by RTTRP) with an unlimited number of trucks and trailers. The authors concluded that significant reductions in total distance are possible by using more trailers. Lin et al. [12] considered the TTRP with time windows (denoted by TTRPTW), and used a simulated annealing heuristic to solve it. Derigs et al. [3] studied the TTRP with time windows. They introduced an extension of the TTRP called the TTRP without load transfer. In many operational scenarios, the commodities being transported are heavy and difficult to load. Therefore, it is useful to alter the model to disallow the transfer of goods between trucks and trailers that are on route to customers.

Drexl [4] introduced a more general variant of vehicle routing problem with trailers and transshipments (VRPTT) that occurred in the collection of raw milk at farm yards. The key difference between the VRPTT and the TTRP lies in the non-fixed truck-trailer assignment. The TTRP can be viewed as a special case of the VRPTT with fixed truck-trailer assignment. This is a much easier problem than the VRPTT, because there are no synchronization requirements other than covering the customers. Due to the fact that all vehicle components must eventually return to the depot, the VRPTT can be seen as a generalization of a variety of vehicle routing problems that involve synchronicity constraints [18]. There are three things in common between the VRPTT and SB-VRP. First, there are special locations for parking, pick-up, or other operations. Second, fixed and variable costs are considered in the objective function, and they are not equal for the trucks and trailers. Third, there are several types of operations at the transshipment locations, such as parking, pick-up, swap, and exchange. Operation time is taken into account. For example, in the VRPTT, operation time depends on the amount of load transferred; in the SB-VRP, it depends on the type of operation.
Drexl [5] introduced the generalized truck and trailer routing problem (GTTRP). The GTTRP is derived from the VRPTW when the truck-trailer assignment is fixed. Drexl solved this problem with branch-and-price and heuristic column generation. The key differences between the GTTRP and the SB-VRP are parking places, operation costs, and time windows. In the GTTRP, parking and load transfer can be done at the trailer customers and at the transshipment locations. These can happen only at swap locations in the SB-VRP. Second, operations at the transshipment locations (parking and load transfer) have no additional cost in the GTTRP [5]. Four types of operations at the swap locations (park, pick-up, swap, exchange) have different costs in the SB-VRP. Customers and transshipment locations can have time windows in the GTTRP. There are no time windows in the SB-VRP.

3. Problem Description

We use a directed graph $G = (N, A)$ for the SB-VRP, where $N = \{D, S, C\}$ denotes the location set, and $A = \{(i, j) \mid i, j \in N, i \neq j\}$ is the arc set. $D = \{0\}$ denotes the depot set and has only one element, because there is only one depot in the SB-VRP. $S = \{1, \ldots, s_n\}$ denotes the swap-location sets where $s_n \in N$ is the number of swap locations. $C = \{s_n + 1, \ldots, s_n + c_n\}$ denotes the customer set where $c_n \in N$ is the number of customers. Each customer $i$ requires quantity $q_i$. A homogeneous fleet of an unlimited number of trucks, semi-trailers, and swap-bodies is located at a single depot. Side loading (or unloading) of a swap-body is possible. This ensures that when a customer is visited by a train, the sequence of swap-bodies on the truck and semi-trailer is not relevant. Only swap-bodies can be loaded, and the maximum load is $Q$. The delivery at each customer takes a specified amount of time $s_{ti}$. At a swap location, a train or truck can make the following four operations (each operation takes a different amount of time), which are illustrated in Figure 4:

- **Park.** A train parks its semi-trailer and continues the route with only the swap-body loaded on the truck bed.
- **Pick up.** A truck picks up the semi-trailer it previously deposited at this location and departs as a train.
- **Swap.** A truck parks the swap-body that it is currently carrying and picks up the other swap-body from the semi-trailer that it had parked there at an earlier stage of the route.
- **Exchange.** A train parks the semi-trailer, exchanges swap-bodies between the truck and the semi-trailer and continues the route with this swap-body as a truck.

All routes start and end at the depot. The distances and driving times between all relevant locations may be asymmetric. A route has a maximum duration which includes driving time, service time of its customers, and operation time at each swap location. The goal is to generate a set of routes with minimum total cost. The total cost ($TC$) of a solution is given by Equation 1.

$$TC = FC + VC$$

where $FC$ is the fixed cost of the vehicle components used in the solution and $VC$ represents the variable costs associated with the routes. The variable cost is proportional to the distance and duration of a route.

A preliminary analysis of solutions to the SB-VRP reveals that there are three types of routes:

- A truck route that uses a truck and only one swap-body.
- A train route that uses a truck and two swap-bodies, and serves no truck-only customers.
- A combined train route that uses a truck and two swap-bodies, and serves some truck-only customers (i.e., there is at least one subtour that starts and ends at a swap location). The rest of the combined train route without the subtour is the main tour.
We point out that a truck-only customer can only be on a truck route or a subtour of a combined train route. A train-only customer can only be on a train route or the main tour of a combined train route.

In Figure 5, we show a SB-VRP solution with the three types of routes. There are two routes: a truck route and a combined train route that has only one subtour. The total cost is the sum of the fixed cost and the variable cost of two trucks and one semi-trailer, and the costs of the swap-actions at the swap location. In this example, the swap-actions are park and pick up.

4. Heuristic Approach

We develop a local search heuristic for the SB-VRP that has three phases. Since the main intuition behind our approach is to first reduce the problem to a traditional VRP (and then use a standard variable neighborhood descent metaheuristic framework to proceed), we refer to this approach as VRP-Reduce. In Table 2, we describe our approach.
Figure 5. An example of a solution to the SB-VRP

Table 2. Three phases of VRP-Reduce

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Initialization. Transform the SB-VRP into a VRP. Apply the simulated annealing procedure in [8] to the VRP.</td>
</tr>
<tr>
<td>2.</td>
<td>Post-processing. Apply a post processor to the VRP solution from Phase 1 to produce a solution to the SB-VRP.</td>
</tr>
<tr>
<td>3.</td>
<td>Improvement procedure. Apply improvement heuristics to the SB-VRP solution.</td>
</tr>
</tbody>
</table>

4.1. Phase 1: Initialization

We use a slightly modified version of the vehicle routing heuristics (VRPH) code library in [8] on the SB-VRP. These heuristics are part of an open source C++ package for the vehicle routing problem that is hosted by COIN-OR (http://www.coin-or.org/projects/VRPH.xml). Since the heuristics in [8] are designed for the standard vehicle routing problem, we modify them to handle the SB-VRP. Since the traditional VRP uses only a homogeneous fleet, we are restricted to a single route type at this phase of the procedure. Therefore, for the SB-VRP, all routes are generated as train routes when all customers are train possible; otherwise, all are truck routes. Two special cases may occur. In the first case, the loads of some train routes are not larger than the swap-body capacity. Therefore, all customers can be served by the truck, and there is no need to use a semi-trailer. In the second case, there are train-only customers that cannot be added to a truck route. The solution produced by the heuristics in [8] would be incomplete, i.e., the train-only customers are not on any truck routes because of the capacity constraint. We deal with these two cases in the post-processing phase. The simulated annealing procedure that we use to generate an initial solution to the SB-VRP is given in Table 3.

We note that there is no fixed cost in the VRP. However, the fixed cost may contribute a significant amount to the total cost of the SB-VRP solution. Using heuristics designed for the VRP may not generate good solutions to the SB-VRP. Taking these considerations into account, we modified the codes in [8], and explain our modifications in the following paragraphs.
Table 3. The simulated annealing procedure

**Input:** A classic VRP instance  
**Output:** $S_0$, a solution to the VRP that may be transformed into a feasible solution for the SB-VRP.  
1: Use the Clarke-Wright savings procedure to generate an initial solution to the VRP.  
2: Apply the local search procedures in [8] sequentially for inter-route improvement:  
   a. one-point move  
   b. two-point move  
   c. 2-opt  
   d. three-point move  
All local search procedures use a neighbor list. Accept the first improvement found and accept worse solutions with a specified probability. Repeat this step twice before cooling (with a cooling ratio of .98). The procedure terminates after 200 cooling iterations.  
3. Output the best-known solution, $S_0$.

We observe that the total cost of a solution to the SB-VRP has four components: (a) the fixed cost associated with using the vehicle, (b) the cost associated with the distance traveled, (c) the cost associated with the travel time; and (d) the cost associated with the service time that is constant in all feasible solutions. We want to develop a solution procedure that minimizes the first three components. To construct the cost matrix, we start with the travel time matrix and add the service time to the corresponding columns (e.g., the first customer’s service time gets added to the second column). Therefore, the solutions generated by [8] satisfy the maximum route duration constraint. It is clear that these solutions minimize component c. Travel time and travel distance are more or less proportional, so that these codes also tend to minimize component b.

Based on our observations and an analysis of the total cost, we add a penalty for using additional routes by specifying a service time at the depot. A cost is incurred each time a new route is constructed or the depot is visited. The amount of service time at a depot (denoted by $st_{depot}$) depends on the problem and type of routes to be constructed and is given by

$$st_{depot} = \alpha \times \frac{\text{fixed cost of a route}}{\text{cost of vehicle per time}}.$$  \[2\]

The fixed cost of a route refers to the vehicle cost. If train routes are constructed, then the fixed cost is the sum of the truck and semi-trailer costs. Otherwise, it is only the fixed cost of a truck. Since the cost matrix reflects time, we need to convert the monetary units of the fixed cost into time units. We use the cost of the vehicle per time unit, which is a specified parameter in the problem.

The parameter $\alpha$ requires explanation. Intuitively, we want to distribute the fixed cost of a route over components b and c, so that we do not change the ratio of component b to component c significantly. For example, suppose the cost associated with travel time is 1,500, the cost associated with the travel distance is 1,000, and the fixed cost of the route is 100. We split the fixed cost of 100 into 60 and 40, so that the ratio of cost to time and cost to distance are the same ($\frac{1500}{1000} = \frac{60}{40}$). Therefore, $\alpha$ is 0.4 in this example.

The value of $\alpha$ can also be derived from problem instances using $\alpha = (1 + \frac{\text{vehicle cost per distance}}{\text{vehicle cost per time}} \times \bar{\nu})^{-1}$, where $\bar{\nu}$ is the average velocity computed from the distance-
time matrix. The derivation is shown in (3) to (6).

\[
\alpha = \frac{\text{cost associated with travel time}}{\text{cost associated with travel time} + \text{cost associated with travel distance}} \quad (3)
\]

\[
= \left[1 + \frac{\text{cost associated with travel distance}}{\text{cost associated with travel time}}\right]^{-1} \quad (4)
\]

\[
= \left[1 + \frac{\text{vehicle cost per distance} \times \text{total distance}}{\text{vehicle cost per time} \times \text{total time}}\right]^{-1} \quad (5)
\]

\[
= \left[1 + \frac{\text{vehicle cost per distance}}{\text{vehicle cost per time} \times V}\right]^{-1} \quad (6)
\]

Since the value of \( V \) is determined only after a route is constructed, we have to estimate its value beforehand. We sum all the distances in the distance-time matrix (excluding those involving swap locations) and divide it by the sum of all the times in the distance-time matrix (excluding those involving swap locations). After the service time at a depot is determined, we increase the maximum route duration by the same amount.

4.2. Post-processing Procedure

In this phase, we handle two special cases to generate a feasible complete solution where all customers are routed. In the first case, we transform the train route into a truck route if its load is no larger than the swap-body capacity. In this case, cost could be reduced, because the number of semi-trailers is reduced. In the second case, we generate a new train route for each unrouted train-only customer. These train routes are combined with the truck routes from the VRP solution to produce a complete solution to the SB-VRP.

4.3. Improvement Procedure

After an initial solution is generated by Phases 1 and 2, a variable neighborhood descent (VND) [9] procedure is used for further solution improvement. VND is shown in Table 4, where \( N_i \) is the maximum number of consecutive iterations with no improvement, and \( N_p \) is the maximum number of tries to find a feasible move for perturbation. The variable \( isPerturb \) shows whether a feasible perturbation move is found.

<table>
<thead>
<tr>
<th>Table 4. The VND procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Complete solution ( S_0 ) generated by Phases 1 and 2;</td>
</tr>
<tr>
<td><strong>Output:</strong> ( S_{\text{best}} )</td>
</tr>
<tr>
<td>1: Let ( S = S_0; S_{\text{best}} = S_0; p = 0; isPerturb = true; )</td>
</tr>
<tr>
<td>2: WHILE (isPerturb is true and ( p &lt; N_i ))</td>
</tr>
<tr>
<td>3: Apply local search defined by Or-opt, 2-opt, swap-single, and swap-string sequentially until no improvement is found; the incumbent solution is denoted by ( S^* );</td>
</tr>
<tr>
<td>4: Let ( isPertrub = false; )</td>
</tr>
<tr>
<td>5: IF ( f(S^*) &lt; f(S_{\text{best}}) )</td>
</tr>
<tr>
<td>6: ( S_{\text{best}} = S^*; p = 0; )</td>
</tr>
<tr>
<td>7: ELSE ( p = p + 1; )</td>
</tr>
<tr>
<td>8: WHILE (isPerturb = false)</td>
</tr>
<tr>
<td>9: IF Perturbation(( S_{\text{best}}, N_p )) is successful</td>
</tr>
<tr>
<td>10: ( isPerturb = true; )</td>
</tr>
<tr>
<td>11: Output ( S_{\text{best}} );</td>
</tr>
</tbody>
</table>
4.3.1. Neighborhood  The neighborhood structure is defined by five types of moves (Or-opt, 2-opt, swap-single, swap-string and truck-only customer migration). The 2-opt procedure is applied to a single route, while the remaining moves are applied to two different routes. If either of the two routes is a train route, these operators are only applied on its main tour.

The Or-opt procedure removes a string of customers from a route and inserts the string on a different route. In order to guide the search in a promising direction, the length of a string is restricted by a parameter that excludes long arcs. The idea is that when we look for strings of locations to relocate to another route, if there’s a leg of the customer string that is greater than a specified amount (the parameter is denoted by consecutiveProximity), then it is not considered in the search. For example, if we set the parameter to 1,000, and a route has arc lengths 500, 600, 800, 2,000, 300, 100 (i.e., the distance from the depot to customer 1 is 500; from customer 1 to customer 2 is 600, etc.), then we consider moving the first three locations (or substrings that involve only the first three customers) to another route, and the last two locations (or substrings that involve only the last two customers) to another route. We would not consider moving any more than the first three customers because that string would contain an arc with a cost of 2,000 which is greater than 1,000. If there is a truck-only customer in the string, and we are moving the customer to the main tour of a train route, then the move is infeasible and is not carried out. In Figure 6, we show a move with Or-opt. The consecutive customers between \( v_i \) and \( v_j \) are removed from route \( R_1 \), and inserted before \( v_k \) on route \( R_2 \). The dotted line indicates there are other customers between \( v_i \) and \( v_j \). It is important to note that this procedure is slightly modified from a standard Or-opt in that we take into account all elements of the objective function, including time, distance, and (most importantly for our case), the fixed vehicle costs incurred when a route is used.

The 2-opt procedure is applied to each single route until no improvements can be made. If the route is a train route and it has a subtour, then the subtour node will be considered as a single dummy node. In Figure 7, we show the 2-opt improvement procedure. The dashed lines in Figure 7(a) indicate the deleted arcs; the dashed-dotted lines in Figure 7(b) are added arcs.

The swap-single and swap-string moves are similar. Swap-single exchanges two customers on the main tours of different routes. Swap-string exchanges a string with less than three
customers on one route with customers on a different route (there is no restriction on the number of customers). Both routes must be truck routes or main tours of train routes. We evaluate all feasible exchanges, and select the exchange with the largest reduction in cost. In order to control the computing time, the procedure stops when the specified maximum number of trials is reached. In Figures 8 and 9, we provide examples of swap-single and swap-string moves, respectively.

The truck-only customer migration procedure tries to move truck-only customers from truck routes into subtours on train routes. Our motivation is that subtours are useful for incorporating truck-only customers into train routes (as opposed to reducing the cost per unit distance of the subtour). For each truck-only customer, we compute the cheapest feasible insertion position for each existing train route. Then, taking this position for the truck-only customer as fixed, we determine the first possible location that a subtour may begin without violating any constraints. The cheapest feasible swap location is used. The start position of the subtour is then moved up, one-by-one, and whenever it becomes feasible, the end position of the specified subtour is also incremented. This procedure produces feasible moves which are ordered by cost. We try each move (beginning with the cheapest) until a new best solution is found. The feasible moves are then recalculated.
4.3.2. Perturbation  In order to escape from a local optimum, we apply a perturbation procedure. We use the swap-string operator to perturb the incumbent solution with no restriction on the number of customers in each string. We check each pair of routes. If a feasible move is found, we make the move regardless of cost.

5. Computational Results

In this section, we test the performance of the VRP-Reduce heuristic on the SB-VRP test instances provided by the VeRolog Solver Challenge 2014. The results are given in the following sections.

5.1. Results on the SB-VRP test instances

The VeRolog Solver Challenge 2014 has nine test instances of the SB-VRP that are divided into small, medium, and large problems. In addition, there are three test instances that are supplied without any information about the problem size. For these three instances, we were only provided the solver’s performance (we cannot discern run time, or intermediate results). For each instance, there are three profiles: all train customers (denoted by all with trailer), all truck-only customers (denoted by all without trailer), and both train customers and truck-only customers (denoted by normal). For each instance, there is only one depot.

Our approach is implemented in C++ and run on a 2012 MacBook Air. Our results are given in Table 5. In this table, we give the solution cost before improvement and after improvement.

We remark that, since this is a new problem, and the full results of the competition have not been publicly released as of fall 2014, it is difficult to determine the quality of our solutions. However, our results placed in the top seven of the final standings (out of the results of the original 27 teams), thereby giving us reasonable confidence in the performance of our algorithm. It is also worth noting that all competitors observed a lack of subtours in the solutions that they generated (relative to heuristically generated solutions to the TTRP, for example). It remains to be seen whether or not this is a feature of the problem, or a feature of the instances provided for the competition.

Finally, we note that our run times are quite reasonable, even for the larger problem sizes, and demonstrate a local search space that, on average (for the normal instances, with both
Table 5. Results by VRPH

<table>
<thead>
<tr>
<th>Instance</th>
<th>n&lt;sub&gt;sl&lt;/sub&gt;</th>
<th>n&lt;sub&gt;cust&lt;/sub&gt;</th>
<th>Cost Before Improvement</th>
<th>Cost After Improvement</th>
<th>% Improvement</th>
<th>CPU (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>small_normal</td>
<td>20</td>
<td>57</td>
<td>5452.99</td>
<td>4959.00</td>
<td>9.96</td>
<td>1.2</td>
</tr>
<tr>
<td>small_all_without_trailer</td>
<td>20</td>
<td>57</td>
<td>5357.03</td>
<td>5356.36</td>
<td>0.01</td>
<td>1.4</td>
</tr>
<tr>
<td>small_all_with_trailer</td>
<td>20</td>
<td>57</td>
<td>4930.06</td>
<td>4873.05</td>
<td>1.17</td>
<td>1.1</td>
</tr>
<tr>
<td>medium_normal</td>
<td>41</td>
<td>206</td>
<td>8735.81</td>
<td>8297.25</td>
<td>5.29</td>
<td>17.3</td>
</tr>
<tr>
<td>medium_all_without_trailer</td>
<td>41</td>
<td>206</td>
<td>8698.13</td>
<td>8628.37</td>
<td>0.81</td>
<td>12.3</td>
</tr>
<tr>
<td>medium_all_with_trailer</td>
<td>41</td>
<td>206</td>
<td>8434.87</td>
<td>8335.55</td>
<td>1.19</td>
<td>15.4</td>
</tr>
<tr>
<td>large_normal</td>
<td>99</td>
<td>548</td>
<td>22851.10</td>
<td>22051.40</td>
<td>3.63</td>
<td>268.9</td>
</tr>
<tr>
<td>large_all_without_trailer</td>
<td>99</td>
<td>548</td>
<td>22593.40</td>
<td>22419.40</td>
<td>0.78</td>
<td>262.5</td>
</tr>
<tr>
<td>large_all_with_trailer</td>
<td>99</td>
<td>548</td>
<td>21460.50</td>
<td>21317.00</td>
<td>0.67</td>
<td>250.3</td>
</tr>
<tr>
<td>new_normal</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>26712.40</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>new_all_without_trailer</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>26712.40</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>new_all_with_trailer</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>26658.10</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

NA = Not available
n<sub>sl</sub> = number of swap locations
n<sub>cust</sub> = number of customers

truck-only and trailer-accessible customers) improves the quality of the solution by 6.3% over treatment as a VRP. This, at the very least, illustrates that there are significant gains to be made from the use of swap locations, despite the lack of subtours.

6. Conclusions

In this paper, we introduced a construction heuristic to solve a new problem in the operations research literature, the Swap-Body Vehicle Routing Problem. In this problem, the objective function is based on parking locations and detachable vehicle components as well as time, distance, and vehicle load. Our heuristic relied on modeling the problem as a traditional VRP variant, and then used variable neighborhood descent to improve the solution. The improvement procedures were more effective on instances that had a full range of constraints rather than on special cases (e.g., instances containing customers with no accessibility restrictions). This suggests that our procedures exploited the special features of the problem. Computational results were reported for the benchmark instances from the 2014 VeroLog Solver Challenge.

In future work, we hope to compare our results to solutions generated by standard bounding procedures (using linear programming-based approaches), as well as to solutions reported by competition participants.

7. Acknowledgments

This paper is supported in part by Humanity and Social Science Youth Foundation of Ministry of Education of China under Grant No. 13YJC630010, and the scholarship from China Scholarship Council under Grant No. 201208120035.

References


