Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Vehicle Routing Problem with Divisible Deliveries and Pickups

Gábor Nagy, Niaz A. Wassan, M. Grazia Speranza, Claudia Archetti

TRANSPORTATION SCIENCE
Vol. 49, No. 2, May 2015, pp. 271–294
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2013.0501

The Vehicle Routing Problem with Divisible Deliveries and Pickups

Gábor Nagy, Niaz A. Wassan
Centre for Logistics and Heuristic Optimisation, Kent Business School, University of Kent, Canterbury CT2 7PE, United Kingdom
{g.nagy@kent.ac.uk, n.a.wassan@kent.ac.uk}

M. Grazia Speranza, Claudia Archetti
Dipartimento Metodi Quantitativi, Università degli Studi di Brescia, 25122 Brescia, Italy
{speranza@eco.unibs.it, archetti@eco.unibs.it}

The vehicle routing problem with divisible deliveries and pickups is a new and interesting model within reverse logistics. Each customer may have a pickup and delivery demand that have to be served with capacitated vehicles. The pickup and the delivery quantities may be served, if beneficial, in two separate visits. The model is placed in the context of other delivery and pickup problems and formulated as a mixed-integer linear programming problem. In this paper, we study the savings that can be achieved by allowing the pickup and delivery quantities to be served separately with respect to the case where the quantities have to be served simultaneously. Both exact and heuristic results are analysed in depth for a better understanding of the problem structure and an average estimation of the savings due to the possibility of serving pickup and delivery quantities separately.

Keywords: vehicle routing; divisible deliveries and pickups; metaheuristics; tabu search
History: Received: June 2012; revisions received: May 2013, July 2013; accepted: September 2013. Published online in *Articles in Advance* November 15, 2013.

1. Introduction

This paper focuses on an extension of the class of vehicle routing problems (VRPs) known as *vehicle routing problems with deliveries and pickups* (VRPDP). The main difference between these problems and the VRP is that customers may *receive* or *send* goods, whereas in the VRP all customers just receive goods from a depot. In the context of these problems, customers who only receive goods are called *delivery* or *linehaul* customers, whereas those only sending goods are called *pickup* or *backhaul* customers—in many applications, however, customers will both send and receive goods. Given customer distances and demands (these include both pickup and delivery demands), we must find a set of routes to minimise the total travelling cost while meeting customer demands. The main *constraint* is that the capacity of the vehicle cannot be exceeded; however, other constraints such as maximum distance or time windows may exist. From a practical point of view, the VRPDP fall within the field of *reverse logistics*, a field that is gaining increasing importance because more people are becoming environmentally conscious. From a mathematical point of view, this problem is an NP-hard *combinatorial optimisation* problem.

We assume that all delivery goods come from depots and all pickup goods are taken to depots. This excludes the possibility of goods travelling directly from one customer to another and implies that delivery goods and pickup goods are not substitutable. In the VRPDP the vehicle may often carry a mixture of delivery and pickup goods: it starts from the depot carrying only delivery goods; at some stage a mixture of goods may occur; finally, the vehicle returns to the depot carrying pickup goods only. At each customer location the load on the vehicle may increase or decrease, resulting in a *fluctuating load*. Hence, checking feasibility must be carried out for each route leg. Thus, the VRPDP is conceptually a harder problem than the VRP, where checking feasibility only needs to be done for the first arc of each route. In fact, one of the main difficulties in solving the VRPDP lies in checking load feasibility.

In this paper, we focus on an interesting, but rarely addressed, model within the VRPDP, called the *VRP with divisible deliveries and pickups* (VRPDDP), comparing and contrasting it to its more common counterpart, the *VRP with simultaneous deliveries and pickups* (VRPSDP). Because these are not well-known VRP variants, and often the terminology used in the subject literature is confusing, we devote the next section to properly defining these models.

The aims of this paper will become clearer once the problem is properly defined. As a quick summary, we are interested in:

1. What characterises instances where the VRPDDP is a more appropriate model than the VRPSDP?
2. What characterises the customers that are treated differently in the VRPDDP as compared to the VRPSDP?
3. What shapes do VRPDDP routes take? (Note that unlike in the VRP, routes may take shapes other than the classical "petals.")

The next section presents a classification of the various VRPDP models and restates our aims in a more precise manner. This is followed by a detailed literature review. Section 4 analyses the savings that can be achieved in the VRPDDP with respect to the VRPSDP. Section 5 presents a mixed-integer linear program (MILP) formulation and an analysis of exact results, whereas §6 analyses results found by a reactive tabu search heuristic. An improved heuristic based on the results of this analysis is given in §7. Finally, we present some conclusions and future research directions in §8.

2. The Vehicle Routing Problem with Divisible Deliveries and Pickups

In the VRPDDP, a set of customers is given, requesting a delivery and/or pickup service. A fleet of homogeneous vehicles, located at a single depot, is available to serve these customers. All delivery goods come from the depot, and all pickup goods are transported to the depot. Each vehicle can transport both pickup and delivery goods and has a maximum capacity limit. Each pickup or delivery request has to be satisfied by a single visit. However, a customer requiring both a pickup and delivery service can be served by two different visits. The objective is to find a set of vehicle routes satisfying the demands of all customers, never exceeding the vehicle capacity, and such that the total distance travelled is minimized.

To better understand the structure of the VRPDDP, we first put it into the context of other pickup and delivery problems. Then, we discuss some research issues and this will enable us to restate the research aims of this paper more precisely.

2.1. Classification of VRPDP

A classification of vehicle routing problems with deliveries and pickups can be given according to the patterns of goods movement, the characteristics of the customers, and restrictions on goods transported on vehicles. Unfortunately, names of problem versions in the literature are often confusing—different authors may use the same term to mean different problems. Although two recent reviews on the VRPDP both present a clear taxonomy, their simultaneous appearance means that the names adopted for the same problem are often different. To help the reader, we will make reference to both taxonomies. For the sake of conciseness, we shall refer to Berbeglia et al. (2007) as *BCGL* and to Parragh, Doerner, and Hartl (2008) as *PDH*.

The first classification is according to the *transport pattern* of goods.

1. In some problems, an item needs to be moved from a customer to another customer. The depot here serves as a base for the vehicles, but these leave the depot empty and return empty. This transport pattern is relevant to dial-a-ride and courier problems. It is called "*transportation between customers* (*VRP with pickup and delivery—VRPPD*)" by PDH. BCGL divides this problem into two classes called "*many-to-many*" and "*one-to-one*" (referred to as "unpaired" and "paired" by PDH). This is in fact a quite different problem from the VRPDP as defined in §1. Because vehicles perform pickups before deliveries, we prefer to refer to this problem as the VRP with pickups and deliveries (VRPPD) to distinguish it from the VRPDP. For a review of the literature, we refer to the surveys mentioned and do not consider this type of problem any further.
2. Our focus is on the transport pattern where all goods must either originate from, or be destined to, a depot. Goods may not be taken directly from one vehicle to another. In these problems, depots serve as hubs or sorting centres. This is typical in mail transportation or where there are two distinct types of goods (e.g., bottled drinks coming from a depot and empty bottles returning there). It is called "*one-to-many-to-one*" by BCGL and "*transportation to/from a depot* (*VRP with backhauling—VRPB*)" by PDH.

The second basis of our taxonomy is the characteristics of the customers.

1. In some problems, customers may either receive or send goods, but not both. These customers are referred to as *linehauls* and *backhauls*, respectively. BCGL refers to this problem class as *single demands*.
2. In other problems, there is at least one customer who wishes to both send and receive goods. BCGL refers to this problem class as *combined demands*.

Third, we may have some restrictions on the travel pattern of the vehicles. One such restriction is that a vehicle may not carry delivery and pickup goods on board at the same time. (Otherwise, the physical design of the vehicles may necessitate having to unload some recently picked-up goods to access delivery goods that are stuck behind them on board, leading to delays. This is known as the *load-shuffling problem*.) The second restriction is that customers may request that when a delivery is made to them the pickup goods are taken away at the same time. (A separate visit for delivery and pickup may be deemed inconvenient.) Clearly, the first restriction is more applicable to the case of single demands and the second to the case of combined demands. This yields the following four classes of the VRPDP.

1. The *vehicle routing problem with backhauling* (VRPB) arises when all customers are either linehaul or backhaul, and delivery and pickup goods cannot be transported together. This implies that each vehicle tour visits linehauls first and then backhauls. PDH calls this the *VRP with clustered backhauls*.

2. The *vehicle routing problem with mixed deliveries and pickups* (VRPMDP) allows linehauls and backhauls to occur in any order on a vehicle tour. PDH calls this the *VRP with mixed linehauls and backhauls*.

3. The *vehicle routing problem with simultaneous deliveries and pickups* (VRPSDP) arises when customers wish to both receive and send goods and specify that the pickup must be taken away at the same time the delivery is made. For this class of problems, the terminology used in this paper is the same as that of PDH.

4. The *vehicle routing problem with divisible deliveries and pickups* (VRPDDP) allows two visits to a customer: one for delivery and another for pickup. A customer with two visits is referred to as a "split" customer. Note that we still assume that all of the delivery to a customer is made in a single visit (and the same for pickup). We also note that literature often—confusingly—includes this problem class in the previous one. (The authors of this paper also referred to this problem as VRPSDP previously. After all, here customers simultaneously have delivery and pickup needs, even if they may be served separately. The recent PDH review suggested the term "divisible," an expression we gratefully adopt, because it points to the essential difference between this problem and the VRPSDP.)

Finally, we note that several articles in the literature, especially those seeking theoretical results, restrict themselves to the case of a single vehicle. This is called the travelling salesman problem with deliveries and pickups (TSPDP); its subproblems are referred to by the abbreviations TSPB, TSPMDP, TSPSDP, and TSPDDP.

2.2. Relationships Between Various VRPDP Versions

It is important to note that the abovementioned problems are not isolated from each other. One particular observation—and a very important one for our research—is that they may sometimes be modelled in terms of another problem.

1. The VRPDDP may be modelled as a VRPMDP by creating two fictitious customers, one purely linehaul and one purely backhaul, co-located at the location of each original customer. Note that this doubles the number of customers that is likely to be detrimental on any solution method (be it exact or heuristic).

2. The VRPSDP cannot be modelled as a VRPMDP because the requirement of simultaneous service may not be satisfied.

3. The VRPMDP may be modelled as a VRPSDP by adding a pickup of zero to each linehaul and a delivery of zero to each backhaul. This does not make the model unduly more complicated. (This also implies that the VRPDDP may be modelled as a VRPSDP.)

4. Although the VRPB and the VRPMDP are totally incompatible with each other, they may both be generalised to a model that is currently gaining recognition, the VRP with restricted mixing of deliveries and pickups (VRPRMDP), in which there is some restriction on having a mixture of delivery and pickup goods on board. Because this is not a well-known version, we point the reader to Tarantilis, Anagnostopoulou, and Repoussis (2012) and Nagy, Wassan, and Salhi (2013) for further information.

5. While the "all-deliveries-before-pickups" assumption is somewhat at odds with the VRPDDP and wholly incompatible with the VRPSDP, the idea of restricted mixing could be applied to the VRPDDP. This is likely to force some customers to be served twice.

2.3. Research Issues in the VRPDDP

A central research issue in the VRPDDP is the shape of vehicle routes. Our terminology and discussion follows, to a large extent, the paper of Gribkovskaia et al. (2007).

1. A *Hamiltonian* route is where all customers are served simultaneously.

2. A *double-path* route begins with a path from the depot traversing all customers belonging to the route making deliveries only, then follows this path in the opposite direction making pickups only. (In such a route, only one customer is served simultaneously, and no delivery and pickup goods are ever carried together.)

3. A *lasso* route consists of three segments: The first contains deliveries only. In the second segment, both deliveries and pickups are made. The third segment follows the path of the first in the reverse direction, satisfying the pickup needs of these customers.

4. A *figure-of-eight* route is similar to a Hamiltonian one, except that a single customer is served twice.

5. A *general* route is one of no predetermined shape. Note that all the previous route shapes assume that customers are not split between routes; if they are, the route shapes are deemed to belong to this category.

The following observations were made by researchers on the VRPDDP.

1. Comparing the best route length for different route shapes for the same problem instance, general is better than lasso, which is better than Hamiltonian, which is better than double path (Gribkovskaia et al. 2007).
2. Although the optimal general solution is better than the optimal lasso one, the special lasso structure allows for faster heuristics. Thus, in practice, better lasso solutions than general ones may be found in the same computing time (Hoff et al. 2009).
3. Relatively few customers are served twice in good-quality solutions. Often only one customer is served twice with a figure-of-eight route shape (Gribkovskaia et al. 2007).
4. Lasso solutions were found to be beneficial in combating the load-shuffling problem, because their structure means that free space is created on the initial deliveries-only route segment (Hoff and Løkketangen 2006).

2.4. Relationship to the Split-Delivery Vehicle Routing Problem

In the VRPDDP more than one visit may be made to a customer. This is similar to another growing topic in the VRP literature, the *split-delivery VRP* (SDVRP), where customers' deliveries may be made in more than one visit. Because in this paper we investigate the similarities between these problems, it may be helpful to summarise some observations on the SDVRP, as follows.

1. The optimal route length of the SDVRP and the optimal number of vehicles may be as little as half of the corresponding VRP. It is suggested that the route-length reduction achievable by splitting is because of the reduction of the number of delivery routes (Archetti, Savelsbergh, and Speranza 2006, 2008).
2. There always exists an optimal solution of the SDVRP where no two routes have more than one customer in common, and the number of split customers is less than the number of routes (Dror and Trudeau 1989; Archetti, Savelsbergh, and Speranza 2006).
3. A central research issue is to find the characteristics of instances where splitting gives significant benefits and the characteristics of the customers that are likely to be split. Dror and Trudeau (1989) observed that high demand is a good predictor for splitting, and customers that are close to the depot have a higher chance to be split. The computational experiments of Archetti, Savelsbergh, and Speranza (2008) suggest that splitting gives the largest benefit when the average customer demand is between 50% and 75% of the vehicle capacity, and the demand variance is small. The experiments do not suggest that customer location is a useful predictor of splitting.

Finally, we note that pickups rarely feature in SDVRP research. Mitra (2005, 2008) allowed splitting in the VRPDDP (see §3 for more details). Mosheiov (1998) modelled the split-delivery VRPMDP by creating d_i fictitious co-located customers each with unit demand for each original customer of demand d_i, resulting in a VRPMDP and a huge increase in problem size. Thangiah, Fergany, and Awan (2007) and Nowak, Ergun, and White (2008, 2009) considered splitting in the VRPPD—as mentioned in §2.1, this is a quite different problem from the VRPDP.

2.5. Research Aims

Having defined the problem properly, we can now restate our research aims more precisely.

1. *What characterises instances where splitting gives significant cost reductions*? This will show in which situations the VRPDDP is applicable. Because the VRPDDP is harder to solve than the VRPSDP, if an instance appears to yield a Hamiltonian solution anyway, it will be easier to solve it straightaway as a VRPSDP. Previous studies on the SDVRP show that cost reductions of up to 50% are possible from the VRP. Would the VRPDDP yield such improvements as compared to the VRPSDP? Would the average demand level be a predictor for cost reductions, as it is in the SDVRP?
2. *What characterises the customers that are being served in more than one visit*? On one hand, the answer to this question will enable the logistics company to focus on these customers and investigate any issues of inconvenience arising out of two visits. On the other hand, it will enable us to design more efficient solution algorithms. As stated before, finding general solutions to the VRPDDP can be time consuming. If we could identify customers that are unlikely to be served twice in good solutions, we could restrict our problem by not allowing splitting for such customers. Because one way of solving the VRPDDP is by converting it into a VRPMDP, if instead of doubling the size of the instance we could just duplicate those customers into fictitious linehauls and backhauls that are likely to be served twice, this would reduce the size of the resulting VRPMDP. We hypothesise that customers' demands are likely to play a part. Do their locations matter? Previous research on the VRPDDP suggests that they do, but studies on the SDVRP suggest that they do not.
3. *What shapes do routes take*? We wish to identify a pattern (if there is any) of split and nonsplit customers on a route. Again, such analysis will enable us to design more efficient algorithms. If route shapes are restricted to some given patterns, we can create algorithms that are simple modifications of VRPSDP methods, and thus faster than general VRPDDP algorithms. Previous studies suggest that lasso and

figure-of-eight route shapes often occur in good VRPDDP solutions.

3. Literature Review

For the sake of conciseness, our review is restricted to the VRPDDP only. For other VRPDP versions, the reader is referred to the two comprehensive surveys of the VRPDP by Berbeglia et al. (2007) and by Parragh, Doerner, and Hartl (2008). The SDVRP is reviewed by Archetti and Speranza (2008, 2012). For a comprehensive introduction to vehicle routing, the reader may wish to consult Toth and Vigo (2002).

Mosheiov (1994) addressed the TSPDDP by converting it into a TSPMDP creating fictitious co-located linehauls and backhauls. He proved that any tour can be made feasible by reinserting the depot into a different edge on the tour. This suggests a simple solution approach: find the optimal TSP tour and insert the depot to the nearest such arc that results in a feasible TSPMDP tour. Optimality, of course, is lost: the nearest such arc may be located very far from the depot. An alternative insertion-based heuristic is also given.

Anily (1996) also decomposed customers with both a pickup and a delivery demand into pairs of customers. However, somewhat surprisingly, she also assumed that all deliveries must be made before pickups, yielding a VRPB model. This forces customers with combined demands to be served twice, unless they happen to be the last linehaul and first backhaul. A region-division scheme called circular regional partitioning is proposed. An assignment problem is solved to connect linehaul and backhaul routes.

Salhi and Nagy (1999) and Nagy and Salhi (2005) modelled the VRPDDP directly. The problem is initially solved as a VRPSDP using a route-first, cluster-second heuristic. The "divisible" aspect is accounted for by a pair of improvement routines called *neck* and *unneck*: the first splits a customer into a linehaul and backhaul entity; the second merges these. Neck inserts the backhaul entity into the best position on the vehicle route, hence creating a figure-of-eight shaped route. (It is noted that disabling these routines solves the VRPSDP.) The improvement heuristic also includes standard VRP routines such as 2-Opt, 3-Opt, shift, exchange, and perturb. There are two more VRPDP-specific routines, *reverse* and *reinsert*. Reversing the direction of a route can reduce load levels; this may enable a subsequent insertion of customers. Reinsert, motivated by the work of Mosheiov (1994), inserts the depot to its best possible position on a route. One variant of the heuristic allows infeasible solutions to occur subject to a penalty proportional to the value of maximum load constraint violation in a strategic oscillation framework. An insertion-based method is also developed for comparison purposes. It models the VRPDDP as a VRPMDP and is based on the concept of inserting more than one backhaul at a time, called cluster insertion. Both methods can also cater for multiple depots.

Halskau, Gribkovskaia, and Myklebost (2001) introduced the concept of *lasso* solutions (described in §2.3). A lasso construction heuristic is proposed for the TSPDDP. It builds a TSP route sequentially (using, e.g., the nearest-neighbour method). Each time a load feasibility violation is encountered, a sufficient number of pickups are removed from the beginning of the route to eliminate the violation. Once all customers are on the route, all removed pickups are served on the return way, in the opposite order of deliveries. This method can be adapted to turn a TSP tour into a TSPDDP tour; one just needs to check the tour arc by arc for feasibility violations. If one is encountered, this idea is used to turn the Hamiltonian tour into a lasso. For the VRPDDP, the authors suggest that a cluster-first, route-second approach is best, because for each cluster a TSPDDP can be solved using these ideas.

Mitra (2005, 2008) allows splitting in the VRPDDP, i.e., both deliveries and pickups may be split into several visits. Mitra (2005) presents a simple construction heuristic; Mitra (2008) a parallel clustering heuristic. The experiments do not show a clear indication for the benefits of splitting.

Hoff and Løkketangen (2006) investigated the TSPDDP with restricted mixing. In their model, a mixture of delivery and pickup goods is only allowed if there is sufficient free space to combat the load-shuffling problem. They suggest that lasso solutions are beneficial for this model, because the load level on the vehicle can decrease on the outbound spoke of the lasso until sufficient free space is available for deliveries and pickups to be carried out simultaneously. Initial solutions are found using the algorithm of Mosheiov (1994); these are then improved using a tabu search method based on the 2-opt operator. The authors found that lasso solutions can be an acceptable compromise between the reduction of route length and the complications due to load shuffling.

Gribkovskaia et al. (2007) discuss various route shapes that may occur in the TSPDDP (our terminology in §2.3 is based to a large extent on this paper). Some theoretical properties of these route shapes are presented. An initial TSP solution is found using nearest neighbour or sweep. This tour is then converted to a number of TSPDDP solutions by removing one of its edges. These solutions follow the TSP tour until the removed edge, then return to the first customer, move to the last customer, then follow the TSP tour backwards until the other side of the removed edge, and finally return along the tour to the depot. Such tours have far too many double visits, and hence a merging procedure is used to eliminate them; each vertex is scanned in turn, and if it can feasibly be served in just one of the directions (forward or backward), then it will be bypassed in the other direction. A shift operator is applied to improve on the best result found. The authors also present a tabu search metaheuristic. This finds an initial TSPSDP solution using Mosheiov's (1994) reinsertion heuristic. The objective function caters for feasibility violation by means of a penalty term. The neighbourhood structure consists of the operators neck and unneck, whereas a reoptimisation procedure based on shift is carried out after each improving move, or periodically. The results show that the best solutions are often non-Hamiltonian; such solutions for most instances contain just one customer who is served twice, in a figure-of-eight shape.

Gribkovskaia, Laporte, and Shyshou (2008) tackled the TSPSDP with selective pickups. In this model all deliveries must be served, but pickups are optional. Each pickup generates a certain revenue; balancing the revenue from these pickups and any detour needed to serve these pickups, forms the objective function of the model. A classical heuristic is given, based on an initial Hamiltonian solution. Each customer may be assigned one of three states: simultaneous delivery and pickup, separate delivery and pickup, delivery only—if a customer's status can be changed from delivery only to simultaneous without creating a feasibility violation, then this is done. Improvement operators include shift, neck, unneck, and "shifting pickups": changing the status of a simultaneous customer to separate, if it helps, turns the status of some other customer from delivery only to simultaneous. The tabu search metaheuristic of Gribkovskaia et al. (2007) is also modified to cater for this model.

Hoff et al. (2009) extend the model of Hoff and Løkketangen (2006) to the case of several vehicles. A tabu search metaheuristic creating lasso solutions is proposed based on the operators shift and swap, and 2-opt as postoptimiser. Infeasible solutions are allowed and attract a penalty. The authors compare the lasso solutions on the one hand to VRPSDP solutions, and on the other hand to general (no predetermined route shape) solutions. The latter are found by converting the VRPDDP to a VRPMDP. This doubles the size of the instance and, the authors observe, slows down the heuristic.

4. A Theoretical Analysis of the VRPDDP

The focus of this section is a worst-case analysis of the VRPDDP with respect to the VRPSDP. We also investigate whether certain properties of the SDVRP hold true also for the VRPDDP. Throughout this analysis, we assume that the triangle inequality holds.

4.1. The VRPDDP vs. the VRPSDP

The VRPDDP is a "relaxation" of the VRPSDP in the sense that each customer requiring both pickup and delivery services can be visited twice. This allows a higher flexibility in the design of the vehicle routes, and thus can decrease the corresponding cost. It is interesting to know by how much the cost can be reduced. The following theorem shows the maximum saving that can be achieved by separating the pickup and delivery services. Let $z(P)$ and $k(P)$ denote the minimum route length and the minimum number of vehicles, respectively, for problem P.

Theorem 1. *The ratio* $z(\text{VRPSDP})/z(\text{VRPDDP}) \leq 2$ *and the bound is tight. Moreover,* $k(\text{VRPSDP})/k(\text{VRPDDP}) \leq 2$ *and the bound is tight.*

Proof. To prove the bound, we will proceed as follows: We will consider an optimal solution (w.r.t. z or k) to the VRPDDP. Starting from this solution, we will construct a feasible solution to the VRPSDP having a route length that is at most the double of $z(\text{VRPDDP})$ and having at most double $k(\text{VRPSDP})$ vehicles.

Consider an optimal solution to the VRPDDP and make a copy of each route. For each pair of identical routes, we make the following modifications. Let the first route visit all customers whose delivery belongs to this route and whose delivery is no less than their pickup (and skip the others). The order in which they will be visited will be the same as the order in which their deliveries were served in the original route. This route is called route of type 1. The other route, called route of type 2, stops at all customers whose pickup belongs to this route and whose pickup is greater than their delivery (and skip the others), again preserving the original ordering. In each visit, both delivery and pickup requests will be served at the same time. (Delete any vehicles containing no customers.) This increases both the total route length and the number of vehicles used by no more than a factor of two.

It remains to show that the corresponding solution is a feasible solution to the VRPSDP. First, each customer is visited only once. (Customers split between routes are served by the route that originally served the larger of the delivery and pickup requests; customers split within a route are likewise served in a single visit.) Second, if we consider the routes of type 1, the delivery requests served are a subset of the delivery requests served by the corresponding route in the solution to the VRPDDP; thus, their sum does not exceed the vehicle capacity. Also, as in routes of type 1, all customers have delivery requests that are greater than or equal to pickup requests, the load on the vehicle is monotonously nonincreasing, and thus the routes are feasible. Likewise, in routes of type 2, the pickup requests served are a subset of the

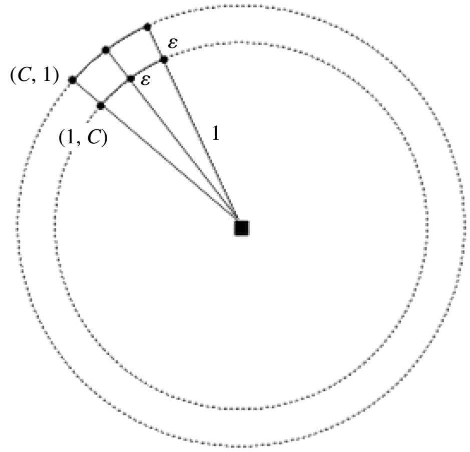

Figure 1 An Illustration of Theorem 1

pickup requests served by the corresponding route in the solution to the VRPDDP. Thus, their sum does not exceed the vehicle capacity and with the load on the vehicle being monotonously increasing, this is sufficient to show feasibility.

To show that the bound is tight, consider an instance where the depot is located in the centre of a circle of radius 1. There are n customers spread out along the circle at a distance ε apart. Furthermore, let there be n additional customers on a circle of radius $1+\varepsilon$, perfectly aligned (along the radius) with the other n customers. Each customer on the inner circle has a pickup demand of 1 and a delivery demand of C, where C is the vehicle capacity. Each customer on the outer circle has a pickup demand of C and a delivery demand of 1. We assume $C \geq 2n$ (see Figure 1). The optimal solution to the VRPSDP visits all customers with out-and-back tours, which results in a cost of $4n+2n\varepsilon$ and requires $2n$ vehicles. On the other hand, an optimal solution to the VRPDDP visits two customers along the radius together, delivering C to the closest customer and picking up C from the farthest customer. The remaining demand, i.e., one pickup unit for all customers on the inner circle and one delivery unit for all customers on the outer circle, is satisfied by one additional route. This results in a cost of $2n+4n\varepsilon+2$ and requires $n+1$ vehicles. Therefore, the ratio between z(VRPSDP) and z(VRPDDP) is equal to $(4n+2n\varepsilon)/(2n+4n\varepsilon+2)$. For n going to infinity and ε going to 0, this ratio tends to 2. Similarly, the ratio between k(VRPSDP) and k(VRPDDP) is equal to $2n/(n+1)$. For n going to infinity, this ratio tends to 2, showing that by allowing splitting, improvements of up to 50% may be achieved. □

4.2. The Concept of Detour Costs

The previous section gave an upper bound on z(VRPDDP) and k(VRPDDP). It is easy to see that a lower bound on these values can be found by solving the corresponding VRP created by setting all pickup values to 0. We define the *detour cost* of the VRPDDP, Δz(VRPDDP), as the increase in route length required to accommodate pickups, i.e., $\Delta z(\text{VRPDDP}) = z(\text{VRPDDP}) - z(\text{VRP})$; similarly for the VRPSDP. Then, $0 \leq \Delta z(\text{VRPDDP}) \leq \Delta z(\text{VRPSDP})$. For the number of vehicles, the corresponding measure is the *additional number of vehicles* Δk(VRPDDP).

There are two benefits of using this measure rather than total route length. First, if $\Delta z(\text{VRPSDP}) = 0$, it is also clear that $\Delta z(\text{VRPDDP}) = 0$, and there is no point in considering splitting. Such a situation arises, e.g., when the pickup of every customer is less than its delivery, because then $z(\text{VRPSDP}) = z(\text{VRP})$. Second, comparing total route lengths may be a misleading measure. If Δz(VRPSDP) is small, then the reduction due to splitting $z(\text{VRPSDP}) - z(\text{VRPDDP})$ will also be small. Using the reduction in detour cost, $(\Delta z(\text{VRPSDP}) - \Delta z(\text{VRPDDP}))/\Delta z(\text{VRPSDP})$, the improvement can be measured in relative terms with respect to the maximum possible gaining, i.e., Δz(VRPSDP), rather than absolute terms. We now show that $(\Delta z(\text{VRPSDP}) - \Delta z(\text{VRPDDP}))/\Delta z(\text{VRPSDP})$ can be equal to 100%; thus, the entire detour cost can be gained by considering divisible deliveries. This is also valid if we consider the reduction in the additional number of vehicles, i.e., $(\Delta k(\text{VRPSDP}) - \Delta k(\text{VRPDDP}))/\Delta k(\text{VRPSDP})$. Consider the following example: Let there be three customers placed on a straight line at distances 1, 2, and 3 from the depot. Customer 1 has delivery of 1 and pickup of 3, customer 2 has delivery and pickup both of 2, customer 3 has delivery of 3 and pickup of 1; vehicles have a capacity of 3. The optimal VRP solution has two vehicles: one visits customers 1 and 2, the other visits customer 3; $z(\text{VRP}) = 10$. The optimal VRPSDP solution has three vehicles, each visiting a single customer; $z(\text{VRPSDP}) = 12$. The optimal VRPDDP solution has two vehicles: one delivers to customer 2 and then visits customer 1, the other visits customer 3 and then picks up from customer 2; $z(\text{VRPDDP}) = 10$. Therefore, both $(\Delta z(\text{VRPSDP}) - \Delta z(\text{VRPDDP}))/\Delta z(\text{VRPSDP})$ and $(\Delta k(\text{VRPSDP}) - \Delta k(\text{VRPDDP}))/\Delta k(\text{VRPSDP})$ equal 1, showing that—in terms of detour costs—allowing splitting can improve the solution by 100%.

4.3. On Properties of the SDVRP and the VRPDDP

Dror and Trudeau (1989) have shown that there always exists an optimal solution of the SDVRP where no two routes have more than one customer in common. This property does not hold for the VRPDDP as shown by the following example. Consider a VRPDDP instance with four customers and vehicle capacity equal to 10. Let the customers be located on a straight line at distances 1, 2, 3, and 4 from the depot. Let the delivery and pickup requests of each customer be the following (the first figure is the delivery request whereas the second is the pickup request): $(10, 1)$ for customer 1, $(4, 5)$ for customer 2, $(5, 4)$ for customer 3, and $(1, 10)$ for customer 4. The optimal solution uses only two vehicles and is the only way to build the following two routes: the first route serves customer 1 completely and then the pickup requests of customers 2 and 3. The second route serves the delivery requests of customers 2 and 3 and then serves customer 4 completely. Thus, customers 2 and 3 are visited by both routes.

Archetti, Savelsbergh, and Speranza (2006) have proved that there exists an optimal SDVRP solution where the number of splits (defined as the number of stops minus the number of customers) is always less than the number of routes. This observation is also not true for the VRPDDP, as shown by the previous example (two splits, two routes). In fact, there is no nontrivial upper limit on the number of splits: in the example of §4.1, every customer is split in the optimal solution.

5. A Mathematical Model and an Analysis of Optimal Results

In this section we provide an integer linear program (ILP) formulation for the VRPDDP and use this formulation to solve some small instances.

5.1. An ILP Formulation for the VRPDDP

As pointed out in §2.2, the VRPDDP can be transformed into a VRPMDP by creating a pair of fictitious co-located customers (one purely linehaul, one purely backhaul) for each customer. The drawback of this approach is that the number of customers is doubled. We have tried to model the VRPDDP directly. However, a flow-based formulation (which is preferable because it has fewer variables than a three-index formulation) could not capture the intricacies of two visits made to a customer. An obvious approach would be just to ensure the total flow from a customer equals the total incoming flow minus delivery plus pickup. However, this would actually allow the vehicle to dump goods at a customer to be picked up later, possibly by another vehicle. More sophisticated ways of accounting for two visits proved to be unable to solve this issue. Hence, we modelled the VRPDDP as a VRPMDP.

Let us introduce the following notation.

Sets

$\mathbf{D} = \{0\}$: the set of depots (consisting of a single depot),

$\mathbf{L} = \{1, 2, \ldots, n\}$: the set of linehaul customers,

$\mathbf{B} = \{n+1, n+2, \ldots, 2n\}$: the set of backhaul customers (backhaul $n+i$ is a copy of linehaul i),

$\mathbf{V} = \mathbf{D} \cup \mathbf{L} \cup \mathbf{B}$: the set of all vertices.

Input variables

d_{ij}: the distance between locations i and j,

q_i: the demand of customer i (this is a delivery demand for $i \in \mathbf{L}$ and a pickup demand for $i \in \mathbf{B}$),

C: vehicle capacity.

Decision variables

x_{ij}: indicator; equals 1 if there is a vehicle travelling from location i to location j; equals 0 otherwise,

R_{ij}: the amount of delivery goods on board on arc ij,

P_{ij}: the amount of pickup goods on board on arc ij.

The VRPDDP can be modelled as follows.

$$\text{Minimise} \quad Z = \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{i \in V} x_{ij} = 1 \quad j \in \mathbf{L} \cup \mathbf{B}, \tag{2}$$

$$\sum_{i \in V} x_{ji} = 1 \quad j \in \mathbf{L} \cup \mathbf{B}, \tag{3}$$

$$\sum_{i \in V} R_{ij} - q_j = \sum_{i \in V} R_{ji} \quad j \in \mathbf{L}, \tag{4}$$

$$\sum_{i \in V} R_{ij} = \sum_{i \in V} R_{ji} \quad j \in \mathbf{B}, \tag{5}$$

$$\sum_{i \in V} P_{ij} = \sum_{i \in V} P_{ji} \quad j \in \mathbf{L}, \tag{6}$$

$$\sum_{i \in V} P_{ij} + q_j = \sum_{i \in V} P_{ji} \quad j \in \mathbf{B}, \tag{7}$$

$$\sum_{i \in L \cup B} P_{0i} = 0, \tag{8}$$

$$\sum_{i \in L \cup B} R_{i0} = 0, \tag{9}$$

$$R_{ij} + P_{ij} \le C x_{ij} \quad i \in \mathbf{V},\ j \in \mathbf{V}, \tag{10}$$

$$x_{ij} \in \{0, 1\} \quad i \in \mathbf{V},\ j \in \mathbf{V}, \tag{11}$$

$$R_{ij} \ge 0 \quad i \in \mathbf{V},\ j \in \mathbf{V}, \tag{12}$$

$$P_{ij} \ge 0 \quad i \in \mathbf{V},\ j \in \mathbf{V}. \tag{13}$$

We present next a brief line-by-line explanation for this formulation.

(1) The objective is to minimise the total distance travelled by the vehicles.

(2)–(3) Every customer is served exactly once.

(4)–(7) Flow conservation constraints. (These constraints also eliminate subtours.)

(8)–(9) Vehicles start with zero pickup load and finish with zero delivery load.

(10) Maximum capacity constraint.

(11)–(13) Set x_{ij} as 0–1 and R_{ij}/P_{ij} as nonnegative variables.

This formulation is based on a two-index VRP formulation. This is achieved by not identifying the vehicle itself as this can be derived from the result. Our proposed formulation requires $(2n+1)^2$ binary variables and $2(2n+1)^2$ continuous variables and is made up of $(2n+1)^2 + 8n + 2$ constraints.

In our experiments, we found the following additional equations/inequalities useful:

$$x_{ii} = 0 \quad i \in \mathbf{V} \quad \text{(no loops)}, \tag{14}$$

$$x_{(n+i)i} = 0 \quad i \in \mathbf{L} \quad \text{(no arc from a backhaul to its corresponding linehaul)}, \tag{15}$$

$$\sum_{i \in L \cup B} x_{0i} \geq \frac{1}{C} \sum_{i \in L} q_i \quad \text{(minimum number of vehicles required)}, \tag{16}$$

$$x_{ij} + x_{ji} \leq 1 \quad i \in \mathbf{L} \cup \mathbf{B},\, j \in \mathbf{L} \cup \mathbf{B} \quad \text{(special case of subtour elimination for sets of two customers)}. \tag{17}$$

5.2. Experiments on Small Instances

Because there are no small-sized VRPDDP instances in the literature, we modified some of the VRP instances of Christofides and Eilon (1969), namely CE22, CE23, CE30, CE30(3), and CE33. The number of customers is one less than the number of the instance, i.e., they consist of 21, 22, 29, 29, and 32 customers. (Two 29-customer instances exist: CE30 has no restriction on the number of vehicles; CE30(3) has the number of vehicles set to the minimum value of 3.) We created VRPDDP instances from the VRP instances as follows. Let the delivery demand of each customer remain the same as its original demand in the VRP instance. Let the pickup of the first customer be q_n and the pickup of the any other customer i be q_{i-1}, where q_i is the original demand of customer i. (We chose this scheme so that our instances can be easily constructed by the reader.) The resulting instances are referred to as CE22P, CE23P, CE30P, CE30(3)P, and CE33P.

These instances were solved using IBM ILOG CPLEX (version 12.5) for the case of both VRPSDP and VRPDDP. To calculate detour costs, the corresponding VRP instances were also solved. The results are shown in Table 1, with details of the routes given in Table 2. CPLEX took between a few seconds and about two hours to solve the VRP and between a few seconds and 13 hours to solve the VRPSDP. However, it was unable to solve the VRPDDP formulation directly except for $n = 21$ (two hours) and $n = 22$ (seven minutes), even when left to run for several days. For $n = 29$ and $n = 32$ we employed the following heuristic to establish bounds. We clustered the customers according to the routes they are on in the VRP and the VRPSDP solutions. (We omit the precise details here.) For an upper bound, we allowed splitting only for some clusters; clearly, a feasible solution to this is a feasible solution to the VRPDDP. For a lower bound, we removed the pickup values of the customers in some clusters. (Just as the VRP solution is a lower bound to the VRPDDP, a solution in which some of the pickups are present and the others removed, is a lower bound.) This procedure yielded the optimal solution for CE30P and tightened the bounds for CE30(3)P (3.6% gap) and CE33P (8.3% gap).

Looking at Table 1, we can immediately see the benefits of allowing splitting. For four out of five instances, some cost reduction was achieved. (Note that these are not specially designed instances.) Although the improvements in terms of total route length may appear rather small (up to about 4%), the reduction in detour costs (which we believe to be a better measure) is significant (between 22% and 93%). On these small instances, the number of vehicles was unchanged except for CE22P (a reduction of one vehicle). With just five instances, we cannot draw any conclusions about which types of instances are more likely to yield significant cost reductions due to splitting.

Looking at the detailed solutions in Table 2, we can see that very few customers (one or two) are split (served twice) in the optimal solutions, in line with the observations of Gribkovskaia et al. (2007). Obviously, such a small study does not allow us to draw wide-ranging conclusions, but we try to give some tentative reasons why these customers ended up being served twice, in the following.

Table 1 **A Comparison of VRPSDP and VRPDDP Results on Small Instances**

Instance	CE22P	CE23P	CE30P	CE30(3)P	CE33P
VRP: z (k)	375 (4)	569 (3)	503 (4)	534 (3)	835 (4)
VRPSDP: z (k)	394 (5)*	597 (3)	545 (4)	578 (3)	844 (4)
VRPSDP: Δz (Δk)	19 (1)	28 (0)	42 (0)	43 (0)	9 (0)
VRPDDP: z (k)	385 (4)	571 (3)	545 (4)	LB = 558 UB = 560 (3)	LB = 835 UB = 842 (4)
VRPDDP: Δz (Δk)	10 (0)	2 (0)	42 (0)	26 (0)	7 (0)
Improvement in z (%)	2.28	4.36	0.00	3.11	0.24
Improvement in Δz (%)	47.37	92.86	0.00	40.91	22.22

Notes. z: total route length, k: number of vehicles, $\Delta z = z - z(\text{VRP})$, $\Delta k = k - k(\text{VRP})$, improvement: $(z(\text{VRPSDP}) - z(\text{VRPDDP}))/z(\mathit{VRPSDP})$ or $(\Delta z(\text{VRPSDP}) - \Delta z(\text{VRPDDP}))/\Delta z(\text{VRPSDP})$, LB: lower bound, UB: upper bound, Δz and improvement calculated w.r.t. upper bound.

*If k is set to 4, the optimal value of z is 400.

Table 2 Detailed Route Configurations for Small Instances

Instance	VRPSDP routes	VRPDDP routes
CE22P	0–9–7–5–2–1–6–0	0–**10D**–8–1–2–5–7–9–**12P**–0
	0–12–19–21–17–0	0–**12D**–15–18–20–17–0
	0–13–11–4–3–8–10–0	0–13–11–4–3–6–**10P**–0
	0–16–20–18–15–14–0	0–16–19–21–14–0
CE23P	0–7–8–5–4–21–18–19–20–22–17–14–15–16–3–2–1–6–0	0–7–21–4–5–8–9–**10P**–13–0
	0–10–13–0	0–**10D**–11–**12P**–0
	0–12–9–11–0	0–18–19–20–22–17–14–15–16–3–2–1–6–**12D**–0
CE30P	0–2–5–4–3–20–0	(Optimal solution contains no split customers, same as the VRPSDP solution shown to the left)
	0–18–15–16–13–7–17–9–14–8–12–11–10–23–0	
	0–19–6–1–24–25–29–27–28–26–0	
	0–21–22–0	
CE30(3)P	0–20–19–23–10–11–12–8–14–9–17–7–13–16–15–18–0	0–20–19–23–10–11–12–8–14–9–17–7–13–16–15–18–0
	0–21–26–28–27–29–24–1–22–0	0–21–26–28–27–29–25–24–**6P**–4–**22P**–0
	0–2–5–4–6–25–3–0	0–**22D**–2–5–1–**6D**–3–0
CE33P	0–4–7–9–8–32–11–12–0	0–4–6–**7D**–9–8–32–11–12–2–0
	0–13–17–25–24–23–20–21–22–19–18–10–6–5–3–0	0–13–17–25–24–23–20–21–22–19–18–10–**7P**–5–3–0
	0–26–27–16–28–29–0	0–26–27–16–28–29–0
	0–30–31–14–15–1–2–0	0–30–31–14–15–1–0

Notes. D: delivery service only, P: pickup service only. Split customers are shown in bold.

—Customer 12 in CE22P was probably split because it is relatively near the depot (11 units, average depot-to-customer distance being 28) and has relatively large demand and pickup (22% and 20% of the vehicle capacity, respectively).

—Customer 10 in CE23P was probably split because it has an extremely large demand and a fairly large pickup (91% and 24% of the vehicle capacity, respectively).

—Customer 12 in CE23P was probably split because it is very near the depot (seven units, average depot-to-customer distance being 45).

—Customer 22 in CE30(3)P was probably split because it is located on the straight line connecting the depot and customer 2 (hence, no detour is needed to serve it) and because it has a very large pickup (33% of the vehicle capacity).

—Customer 7 in CE33P was probably split because it is located in a dense cluster of customers (there are five other customers within a radius of five units, including one just one unit away).

—However, no obvious reasons spring to mind regarding customer 10 in CE22P or customer 6 in CE30(3)P.

Finally, we note that the lasso route shape encountered by Halskau, Gribkovskaia, and Myklebost (2001) is not present in our solutions: every split customer is served by a different route for delivery and pickup.

6. Computational Analysis of Heuristic Results

We carried out our computational analysis on a well-known data set and its variants, focusing on the three research aims set in §2.5. The next section explains and justifies our methodology and §6.2 introduces our experiments. Sections 6.3–6.5 focus on each of the research questions in turn.

6.1. Methodology and Justification

To compare the VRPSDP and the VRPDDP we proceed as follows:

Step 1. Solve the VRPSDP using a good-quality metaheuristic.

Step 2. For each genuine customer, create two fictitious customers, one purely delivery, the other purely pickup.

Step 3. Starting from the solution found in Step 1, solve the resulting VRPMDP using a good-quality metaheuristic.

This is a valid approach, as discussed in §2.2, and in line with previous studies, see, e.g., Mosheiov (1994), Salhi and Nagy (1999), and Hoff et al. (2009). The approach has the drawback of having twice as many customers; however, our aim here is to analyse split solutions with the view of creating more efficient solution algorithms.

The prior analysis can be carried out by any good method capable of solving the VRPSDP and the VRPMDP. We chose the reactive tabu search (RTS) metaheuristic of Wassan, Wassan, and Nagy (2008) in Step 1 and the RTS method of Wassan, Nagy, and Ahmadi (2008) in Step 3. Our choice can be justified as follows:

—RTS in general is known to be a very efficient metaheuristic (see Battiti, Brunato, and Mascia 2008).

—The algorithms of Wassan (2007); Wassan, Wassan, and Nagy (2008); and Wassan, Nagy, and Ahmadi (2008) give competitive solutions to the VRPB, VRPSDP, and the VRPMDP, respectively.

—Nagy, Wassan, and Salhi (2013) adapted the Wassan, Nagy, and Ahmadi (2008) method to solve an extension to the VRPMDP. On small instances, the optimality gap was around 1%.

—On the instances of §5.2, RTS has always found the optimal solution or the same upper bound as CPLEX.

For the sake of conciseness, we do not present technical details of the two metaheuristics used here.

6.2. Computational Experiments

We chose one of the most commonly used sets of VRPSDP test instances, namely, that proposed by Salhi and Nagy (1999). This set originally contains 28 instances, ranging from 50 to 199 customers. Distances are Euclidean, and, to eliminate any problem associated with computer precision, are rounded to the nearest integer. Note that instances 6–10, 13, and 14 have a maximum time constraint, whereas instances 11–14 are clustered. A particular characteristic of this data set is that in some instances there are pairs of customers located at the same coordinates. In instances 4 and 9, customer pairs at the same location are: 80 and 150, and 99 and 104. In instances 5 and 10, customer pairs at the same locations are: 3 and 158, 4 and 155, 10 and 189, 58 and 152, 80 and 150, 92 and 151, 99 and 104, and 138 and 154.

Our initial experimentation did not show significant benefits of splitting; hence, we devised further instances. Although various sets of instances were tested, for the sake of conciseness and simplicity, here we report on only two in detail. First, we noticed the average demand and pickup values are very small in the Salhi and Nagy (1999) data set, on average 4% of the vehicle capacity and none larger than 22% of the vehicle capacity, leading to a few long routes. Therefore, we kept the locations of the Salhi and Nagy (1999) data set, but changed the delivery and pickup values by multiplying all values by four and adding $0.1C$. This new data set has delivery and pickup values between 10% and 98% of the vehicle capacity, averaging 26%, leading to many short routes. Second, noting the example in §4 that gave a 50% cost improvement, where the difference between delivery and pickup figures was large, we created such a data set. We added $0.75C$ to the delivery and $0.2C$ to the pickup of every odd customer, and $0.2C$ to the delivery and $0.75C$ to the pickup of every even customer. Coordinates were retained. Thus, for every customer either the delivery or the pickup value is between 75% and 97% of the vehicle capacity, whereas the other value is between 20% and 42%. This means we expect several very short routes. Thus, we finally created the following set of three instances:

—*Set* 1: the original instances by Salhi and Nagy (1999).

—*Set* 2: delivery and pickup values between 10% and 98% of the vehicle capacity.

—*Set* 3: instances with delivery or pickup value between 75% and 97% of the vehicle capacity.

The RTS algorithm was implemented in Fortran 90 and the experiments executed on a Sun-Fire-V440 with UltraSPARC-IIIi Processor, CPU speed 1062 MHz, running Solaris 9. The total number of iterations was set to 1,500 for all instances. All 28 instances of a set were solved in about an hour, which is approximately two minutes per instance (ranging from less than half a minute to about six minutes). A brief analysis of computing times is given in §7.

Tables 3 and 4 compare the simultaneous (Hamiltonian) and divisible (general) solutions for each instance. The former were found using the RTS algorithm of Wassan, Wassan, and Nagy (2008), with exactly the same parameters as above. Because these were taken as the initial solution to our RTS algorithm, the VRPDDP solution will never be worse than the corresponding VRPSDP solution. The VRP solutions necessary for calculating detour costs in Table 4 were again found using the RTS algorithm of Wassan, Wassan, and Nagy (2008), with zero pickup values. Table 5 presents detailed VRPDDP solutions for the instances with a positive improvement. These will be analysed in §6.5 with regard to route shapes. It shows that even in solutions where splitting gives an improvement, only a minority of the customers are served twice. All customers are tabulated in Table 6 and analysed (see §6.4) with the aim of finding out why they were split.

6.3. What Characterises Instances Where Splitting Gives Significant Cost Reductions?

Comparing VRPDDP to VRPSDP solutions (see Table 3), on the instances of Set 1 more than one-third (10 of 28, 36%) of the instances experienced some cost reduction, although the average reduction was only 0.16% (maximum 1.32%, on CMT2X). The number of vehicles was never reduced. We think this is explained by the delivery and pickup figures being too small, thus reducing the need for splitting.

Looking at the results of the instances of Set 2, the situation changes. Route length, on average, is reduced by 1.93% (maximum 6.16%) and the number of vehicles by 3.12% (maximum 8.57%). For every instance, the route length was reduced; for 18 of 28 instances, the number of vehicles was also reduced. This already shows that the savings achievable by splitting are significant. (We note that in this data set all delivery and pickup values are $\geq 0.1\,C$. On a very similar data set, not reported here in detail for the sake of brevity, where the range for deliveries and pickups was between 0% and 88% (rather than 10%–98%), the average saving by splitting was

Table 3 Comparison of VRPSDP and VRPDDP Results Based on Total Route Length

		Set 1						Set 2						Set 3					
		VRPSDP		VRPDDP		Improvement (%)		VRPSDP		VRPDDP		Improvement (%)		VRPSDP		VRPDDP		Improvement (%)	
Instance	Size	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*	*z*	*k*
CMT1X	50	478	3	478	3	0.00	0.00	1,185	19	1,169	19	1.35	0.00	2,315	45	1,943	35	16.07	22.22
CMT1Y	50	476	3	476	3	0.00	0.00	1,101	18	1,093	18	0.73	0.00	2,245	45	1,968	35	12.34	22.22
CMT2X	75	713	7	712	7	0.14	0.00	2,074	38	1,978	36	4.63	5.26	3,586	74	3,217	60	10.29	18.92
CMT2Y	75	694	7	694	7	0.00	0.00	2,028	35	1,903	32	6.16	8.57	3,586	74	3,292	63	8.20	14.86
CMT3X	100	727	5	726	5	0.14	0.00	2,055	32	2,009	30	2.24	6.25	4,210	80	3,805	69	9.62	13.75
CMT3Y	100	723	5	723	5	0.00	0.00	1,884	28	1,860	28	1.27	0.00	4,146	80	3,728	64	10.08	20.00
CMT4X	150	901	8	901	8	0.00	0.00	2,884	47	2,806	45	2.70	4.26	6,263	121	5,738	105	8.38	13.22
CMT4Y	150	859	7	859	7	0.00	0.00	2,612	42	2,605	42	0.27	0.00	6,017	116	5,328	97	11.45	16.38
CMT5X	199	1,090	11	1,083	11	0.65	0.00	3,865	68	3,843	68	0.57	0.00	8,366	167	7,554	147	9.71	11.98
CMT5Y	199	1,053	10	1,052	10	0.10	0.00	3,403	59	3,371	58	0.94	1.69	8,080	163	7,391	142	8.53	12.88
CMT6X	50	555	6	555	6	0.00	0.00	1,185	19	1,169	19	1.35	0.00	2,315	45	1,943	35	16.07	22.22
CMT6Y	50	556	6	556	6	0.00	0.00	1,101	18	1,093	18	0.73	0.00	2,245	45	1,968	35	12.34	22.22
CMT7X	75	899	11	899	11	0.00	0.00	2,083	38	2,027	36	2.69	5.26	3,586	74	3,217	60	10.29	18.92
CMT7Y	75	902	11	902	11	0.00	0.00	2,028	35	1,903	32	6.16	8.57	3,586	74	3,292	63	8.20	14.86
CMT8X	100	874	9	874	9	0.00	0.00	2,055	32	2,009	30	2.24	6.25	4,210	80	3,805	69	9.62	13.75
CMT8Y	100	867	9	867	9	0.00	0.00	1,884	28	1,860	28	1.27	0.00	4,146	80	3,728	64	10.08	20.00
CMT9X	150	1,200	15	1,193	15	0.59	0.00	2,884	47	2,806	45	2.70	4.26	6,263	121	5,738	105	8.38	13.22
CMT9Y	150	1,215	15	1,215	15	0.00	0.00	2,612	42	2,605	42	0.27	0.00	6,017	116	5,328	97	11.45	16.38
CMT10X	199	1,439	19	1,438	19	0.07	0.00	3,865	68	3,843	68	0.57	0.00	8,366	167	7,554	147	9.71	11.98
CMT10Y	199	1,467	19	1,452	19	1.03	0.00	3,403	59	3,371	58	0.94	1.69	8,080	163	7,391	142	8.53	12.88
CMT11X	120	1,009	5	1,009	5	0.00	0.00	3,941	32	3,894	31	1.19	3.13	10,024	90	8,608	77	14.13	14.44
CMT11Y	120	905	4	905	4	0.00	0.00	3,333	29	3,309	28	0.72	3.45	9,727	89	8,372	75	13.93	15.73
CMT12X	100	680	6	680	6	0.00	0.00	2,609	37	2,535	34	2.84	8.11	5,328	83	4,523	68	15.11	18.07
CMT12Y	100	632	5	632	5	0.00	0.00	2,289	33	2,233	32	2.45	3.03	5,114	80	4,304	68	15.84	15.00
CMT13X	120	1,647	11	1,644	11	0.18	0.00	3,941	32	3,894	31	1.19	3.13	10,024	90	8,608	77	14.13	14.44
CMT13Y	120	1,710	12	1,708	12	0.12	0.00	3,333	29	3,309	28	0.72	3.45	9,727	89	8,372	75	13.93	15.73
CMT14X	100	842	10	831	10	1.32	0.00	2,609	37	2,535	34	2.84	8.11	5,328	83	4,523	68	15.11	18.07
CMT14Y	100	854	11	854	11	0.00	0.00	2,289	33	2,233	32	2.45	3.03	5,114	80	4,304	68	15.84	15.00
Average						0.16	0.00					1.93	3.12					11.69	16.41

Notes. z: total route length, k: number of vehicles, improvement: $(z(\text{VRPSDP}) - z(\text{VRPDDP}))/z(\text{VRPSDP})$ or $(k(\text{VRPSDP}) - k(\text{VRPDDP}))/k(\text{VRPSDP})$.

only 0.60%. This shows that the absence of very small deliveries and pickups is a significant factor for splitting to be useful.)

The best results were achieved on the instances of Set 3: an average route-length reduction of 11.69% and an average vehicle number reduction of 16.41% were achieved. (Maximum values were 16.07% and 22.22%, respectively.) This is a very significant saving, especially when compared to the theoretical limit of 50%. However, it is unlikely that such instances occur in realistic situations. Already in this instance set, most (78%) VRPSDP routes contain only a single customer.

It does not appear that the presence of a maximum time constraint is a predictor of splitting. The reduction in the number of vehicles was the same for constrained and nonconstrained instances on all three data sets. The difference in average reduction of route length was insignificant. However, one should expect that if there are very tight maximum time constraints applied, then splitting is unlikely to be beneficial, because vehicles will not be filled to capacity anyway.

There is some evidence that splitting gives more benefit to clustered instances. On the instances of Set 2, where there is a large variation in delivery and pickup figures, and there are many short routes, the reduction in the number of routes is 4.43% for the clustered instances (as opposed to only 2.60% for the nonclustered instances). On the instances of Set 3, where customers have a large imbalance between their delivery and pickup, the reduction in route length is 14.75% for the clustered instances (as opposed to 10.47% for the nonclustered instances). This is sensible, because in clustered instances the intercustomer distances, and hence the detour lengths required to serve a customer twice, are small.

Using *detour costs*, rather than total route length, puts a sharper focus on the improvements (see Table 4). It transpires straightaway that in the instances of Set 1 one reason why there was often no improvement from splitting is that the detour cost Δz was already zero. Of the 28 instances, eight have zero detour cost, and thus no improvement from splitting was to be expected. An average improvement in the detour costs of 17.57% is observed (including

Table 4 Comparison of VRPSDP and VRPDDP Results Based on Detour Cost

		Set 1						Set 2						Set 3					
		VRPSDP		VRPDDP		Improvement (%)		VRPSDP		VRPDDP		Improvement (%)		VRPSDP		VRPDDP		Improvement (%)	
Instance	Size	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk	Δz	Δk
CMT1X	50	7	0	7	0	0.00	—	104	0	88	0	15.38	—	601	13	229	3	61.90	76.92
CMT1Y	50	0	0	0	0	—	—	150	3	142	3	5.33	0.00	592	14	315	4	46.79	71.43
CMT2X	75	36	1	35	1	2.78	0.00	221	2	125	0	43.44	100.00	1,011	25	642	11	36.50	56.00
CMT2Y	75	40	1	40	1	0.00	0.00	273	4	148	1	45.79	75.00	997	26	703	15	29.49	42.31
CMT3X	100	1	0	0	0	100.00	—	290	3	244	1	15.86	66.67	1,033	21	628	10	39.21	52.38
CMT3Y	100	0	0	0	0	—	—	176	1	152	1	13.64	0.00	1,026	23	608	7	40.74	69.57
CMT4X	150	27	1	27	1	0.00	0.00	414	2	336	0	18.84	100.00	1,585	31	1,060	15	33.12	51.61
CMT4Y	150	12	1	12	1	0.00	0.00	314	4	307	4	2.23	0.00	1,396	28	707	9	49.36	67.86
CMT5X	199	42	1	35	1	16.67	0.00	505	6	483	6	4.36	0.00	2,138	45	1,326	25	37.98	44.44
CMT5Y	199	91	2	90	2	1.10	0.00	359	6	327	5	8.91	16.67	1,973	45	1,284	24	34.92	46.67
CMT6X	50	0	0	0	0	—	—	104	0	88	0	15.38	—	601	13	229	3	61.90	76.92
CMT6Y	50	0	0	0	0	—	—	154	3	146	3	5.19	0.00	592	14	315	4	46.79	71.43
CMT7X	75	1	0	1	0	0.00	—	222	2	166	0	25.23	100.00	1,011	25	642	11	36.50	56.00
CMT7Y	75	4	0	4	0	0.00	—	273	4	148	1	45.79	75.00	997	26	703	15	29.49	42.31
CMT8X	100	5	0	5	0	0.00	—	290	3	244	1	15.86	66.67	1,033	21	628	10	39.21	52.38
CMT8Y	100	0	0	0	0	—	—	176	1	152	1	13.64	0.00	1,026	23	608	7	40.74	69.57
CMT9X	150	29	1	22	1	24.14	0.00	408	3	330	1	19.12	66.67	1,585	31	1,060	15	33.12	51.61
CMT9Y	150	0	0	0	0	—	—	315	3	308	3	2.22	0.00	1,396	28	707	9	49.36	67.86
CMT10X	199	19	1	18	1	5.26		505	6	483	6	4.36	0.00	2,138	45	1,326	25	37.98	44.44
CMT10Y	199	56	1	41	1	26.79	0.00	359	6	327	5	8.91	16.67	1,973	45	1,284	24	34.92	46.67
CMT11X	120	56	1	56	1	0.00	0.00	866	7	819	6	5.43	14.29	2,783	23	1,367	10	50.88	56.52
CMT11Y	120	0	0	0	0	—	—	227	3	203	2	10.57	33.33	2,433	21	1,078	7	55.69	66.67
CMT12X	100	45	1	45	1	0.00	0.00	645	6	571	3	11.47	50.00	1,677	22	872	7	48.00	68.18
CMT12Y	100	3	0	3	0	—	—	157	3	101	2	35.67	33.33	1,396	19	586	7	58.02	63.16
CMT13X	120	13	0	0	0	100.00	—	866	7	819	6	5.43	14.29	2,783	23	1,367	10	50.88	56.52
CMT13Y	120	12	1	10	1	16.67	0.00	227	3	203	2	10.57	33.33	2,433	21	1,078	7	55.69	66.67
CMT14X	100	19	0	8	0	57.89	—	645	6	571	3	11.47	50.00	1,677	22	872	7	48.00	68.18
CMT14Y	100	23	1	23	1	0.00	0.00	157	3	101	2	35.67	33.33	1,396	19	586	7	58.02	63.16
Average						17.57	0.00					16.28	34.52					44.47	59.55

Notes. Δz: detour cost, Δk: additional number of vehicles, improvement: $(\Delta z(VRPSDP) - \Delta z(VRPDDP))/\Delta z(VRPSDP)$ or $(\Delta k(VRPSDP) - \Delta k(VRPDDP))/\Delta k(VRPSDP)$.

two instances with an improvement of 100%). Because Δk, the *additional number of vehicles*, was in most cases 0 or 1, it is not surprising that no reduction in the number of vehicles was achieved by splitting. On the instances of Set 2, an average improvement of 16.28% is observed. In line with the observations of Archetti, Savelsbergh, and Speranza (2008), this reduction is likely to have been caused by a reduction in the number of vehicles: the average reduction in Δk is 34.52%. Finally, on the instances of Set 3, average improvements of 44.47% in Δz and 59.55% in Δk are achieved when splitting is allowed. These large values show very clearly the benefits of splitting.

6.4. What Characterises the Customers That Are Being Served in More Than One Visit?

We hypothesised that the customers who are served separately for delivery and pickup may have one or more of the following characteristics: being near the depot, having a large demand or pickup, or being located in a densely populated area. (Our analysis here is based only on the original Salhi and Nagy 1999 instances, because in the additional instances, too many customers were split for a meaningful analysis. In the extreme case of the example of §4.1, every customer is served twice, and the analysis is trivial.) Table 5 lists these characteristics for each of the 61 split customers. For each instance, the nearest 25% of customers to the depot were classified as "near depot" and the 25% of customers with the largest demand as "large demand" (similarly for "large pickup"). The final column classifies the customer as part of a cluster. A customer is considered to be in a cluster if it has at least five neighbours, where a neighbour is defined as a customer that is within a distance of 10% of the average depot-to-customer distance for that instance.

We found that being *near the depot* is the most important characteristic: about four-fifths (48 of 61, 79%) of split customers exhibit this characteristic. This was expected, because it is easy to insert a near-depot delivery to the beginning of a route or a near-depot pickup at the end of a route without greatly increasing the total distance travelled.

Table 5 Characteristics of Split Customers

Instance	Customer	Near depot	Large demand	Large pickup	Cluster
CMT2X	67	Yes	Yes	No	No
	75	Yes	Yes	No	No
CMT3X	30	No	Yes	No	No
CMT5X	*10*	No	No	No	No
	27	Yes	Yes	No	No
	28	Yes	Yes	No	No
	53	Yes	No	No	No
	60	No	No	No	No
	80	No	No	No	No
	111	Yes	Yes	No	No
	112	Yes	Yes	No	No
	138	Yes	No	No	No
	146	Yes	No	No	No
	150	No	No	No	No
	154	Yes	Yes	No	No
	156	Yes	Yes	No	No
	167	Yes	Yes	No	No
	189	No	No	No	No
	196	Yes	Yes	No	Yes
CMT5Y	*10*	No	No	No	No
	28	Yes	No	Yes	No
	68	No	Yes	Yes	No
	156	Yes	No	Yes	No
	190	Yes	Yes	Yes	No
CMT9X	1	Yes	No	No	No
	28	Yes	Yes	No	Yes
	53	Yes	No	No	Yes
	96	Yes	No	No	Yes
	104	Yes	No	No	Yes
	111	Yes	Yes	No	Yes
	138	Yes	No	No	Yes
	146	Yes	No	No	No
CMT10X	28	Yes	Yes	No	No
	53	Yes	No	No	Yes
	80	No	No	No	Yes
	92	No	No	No	Yes
	111	Yes	Yes	No	No
	152	Yes	Yes	No	Yes
	156	Yes	Yes	No	Yes
	196	Yes	Yes	No	Yes
CMT10Y	28	Yes	No	Yes	No
	53	Yes	Yes	No	Yes
	69	Yes	No	No	Yes
	96	Yes	No	No	Yes
	104	Yes	No	No	Yes
	111	Yes	Yes	No	No
	125	No	No	Yes	Yes
	138	Yes	No	No	No
	154	Yes	No	Yes	No
	199	No	No	No	Yes
CMT13X	89	Yes	No	No	Yes
	99	Yes	No	No	Yes
CMT13Y	39	No	No	No	Yes
	87	Yes	Yes	Yes	Yes
	90	Yes	No	Yes	Yes
	91	Yes	No	No	Yes
	92	Yes	Yes	No	Yes
	94	Yes	No	No	Yes
	97	Yes	Yes	No	Yes
	105	Yes	No	No	Yes
CMT14X	43	Yes	No	No	Yes

Having a *high demand or pickup* is also important: about half (31 out of 61, 51%) of split customers have a high demand or pickup. Because load feasibility is the major constraint in our problem, such large customers are the most difficult to place on a route. Hence, splitting them gives additional flexibility, and thus leads to better solutions.

Being *located in a densely populated area* has also proved to be a predictor for splitting: about half (29 of 61, 48%) of split customers have at least five other customers nearby. This makes sense because in dense clusters, making a detour to serve a split customer yields only a small increase in route length.

Our hypothesis explained the occurrence of splitting for most (55 out of 61, 90%) split customers. We then looked at the remaining six to see if any other factor existed contributing to their splitting. For five of them, we found that the reason they are served twice is that they are co-located with another customer. For a pair of co-located customers it makes sense to first deliver to them both, then carry out the two pickups, resulting in one or both of them being split. For easier visualisation, such co-located customers are highlighted in italics in Table 5. This is a particular characteristic of the data set, but if in practice such co-located customers exist, then they are certainly good candidates for splitting.

Only one split customer (60 in CMT5X) is not explained by any of the previous reasons. Therefore, a promising avenue for further research would be to consider splitting only for customers that exhibit one of the abovementioned characteristics.

6.5. What Shapes Do Routes Take?

On the instances where splitting gives significant benefits, the routes contain too few customers for a meaningful analysis. Hence, in this section, again we focus on the more realistic instances of Set 1. Table 6 presents all 119 routes on the instances where splitting occurred. The second column shows the number of split customers on a route, whereas the third describes the shape of the route. We note that slightly more than half the routes (68 out of 119, 57%) contain one or more split customers. From now on, we look only at these routes.

In line with expectations, nearly two-thirds (44 of 68, 65%) of routes contain just one split customer, with very few (4 of 68, 6%) containing more than three.

Most routes (56 of 68, 82%) are in the shape of a simple *cycle*, denoted by "*C*" in Table 3. (To avoid confusion with the terminology of Gribkovskaia et al. 2007, we do not refer to such routes as Hamiltonian; this term is reserved for routes where every customer receives simultaneous service.) Having taken a closer look at the remaining 12 routes, we saw that the issue of co-located customers, a characteristic of

Table 6 Detailed Configurations for Routes with Split Solutions

Instance	*t*	S	Route
CMT2X	1	C	0–**75P**–68–2–62–73–1–43–41–42–64–22–61–28–74–0
	1	C	0–**75D**–30–48–21–47–36–69–71–60–70–20–37–5–0
	1	C	0–**67P**–34–52–27–13–54–57–15–29–45–4–0
	1	C	0–**67D**–35–14–59–19–8–46–0
CMT3X	1	C	0–27–69–1–50–33–81–9–51–**30D**–32–90–63–10–62–19–11–64–49–36–47–46–45–17–84–5–60–89–0
	1	C	0–28–76–77–3–79–78–34–35–71–65–66–20–**30P**–70–31–88–7–48–82–8–83–18–52–0
CMT5X	4	Y	0–132–69–162–101–70–30–20–188–66–128–160–131–32–181–63–126–90–108–***189D***–***10D***–***189P***–***10P***–31–**167P**–**146D**–0
	1	C	0–**111P**–50–102–157–9–135–35–136–65–71–161–103–51–122–1–0
	3	C	0–**27P**–176–33–81–120–164–34–78–169–129–79–185–**196P**–184–**28P**–0
	5	Y	0–**111D**–76–**196D**–116–77–3–158–121–29–24–134–163–68–***80D***–***150D***–***80P***–***150P***–177–109–12–**154P**–0
	1	C	0–**53D**–105–180–198–110–155–4–139–187–39–67–170–25–55–165–130–54–179–149–26–0
	4	Y	0–**28D**–***138P***–***154D***–***138D***–195–21–72–197–56–186–23–75–133–22–41–145–171–74–73–40–**53P**–0
	1	C	0–**112D**–183–94–95–117–97–87–172–43–15–57–178–115–2–58–152–0
	1	C	0–137–144–42–142–14–38–140–44–119–192–91–61–85–93–59–104–99–96–6–**112P**–0
	2	C	0–**156P**–13–151–92–37–98–100–193–191–141–16–86–113–17–173–84–5–118–**60P**–166–89–0
	3	C	0–**156D**–147–**60D**–83–199–125–45–174–46–36–143–49–64–107–123–182–7–194–106–153–52–**146P**–0
	2	C	0–**27D**–**167D**–127–190–88–148–62–159–11–175–19–168–47–124–48–82–8–114–18–0
CMT5Y	1	C	0–27–176–1–122–51–103–161–71–65–136–35–135–9–120–185–77–196–76–**28P**–0
	1	C	0–167–127–**190D**–88–148–62–159–90–126–63–181–32–131–160–128–66–188–20–30–70–101–162–69–132–0
	2	Y	0–153–106–194–7–82–48–47–36–143–49–64–107–175–11–108–***10D***–*189*–***10P***–31–**190P**–146–0
	1	C	0–52–182–123–19–168–124–46–174–8–114–125–45–199–83–18–166–89–**156D**–0
	1	C	0–**156P**–13–117–97–87–42–43–15–57–178–2–115–145–41–22–133–75–74–171–73–152–58–0
	2	C	0–53–105–180–198–110–155–25–55–165–130–54–134–163–24–29–121–**68P**–116–184–**28D**–0
	1	C	0–111–50–102–157–33–81–164–34–78–169–129–79–3–158–**68D**–150–80–177–109–12–138–154–0
CMT9X	1	C	0–27–127–31–10–108–90–63–126–107–19–123–**146P**–0
	1	C	0–69–101–70–30–32–131–128–66–20–122–**1P**–0
	1	C	0–9–13–35–136–65–71–103–51–**1D**–132–0
	2	C	0–**111P**–50–102–33–81–120–34–78–129–79–3–77–**28D**–0
	3	C	0–**111D**–76–116–68–121–29–24–134–150–80–12–**138P**–**28P**–0
CMT9Y	1	C	0–**138D**–109–54–130–55–25–149–26–0
	1	C	0–**53D**–110–4–139–39–67–23–56–75–72–21–0
	1	C	0–105–40–73–74–133–22–41–145–115–2–58–**53P**–0
	2	Y	0–**96D**–***104D***–*99*–***104P***–6–0
	1	C	0–**96P**–59–93–85–61–17–45–125–83–60–118–89–0
	1	C	0–**146D**–52–106–7–82–48–124–46–8–114–18–0
CMT10X	2	**L**	0–**28D**–76–**196D**–77–3–158–29–121–68–116–184–**28P**–0
	1	C	0–**156D**–112–183–96–99–104–93–85–61–173–5–147–0
	1	Y	0–94–***92D***–*151*–***92P***–98–91–16–86–141–191–193–59–6–0
	1	C	0–**156P**–13–87–172–42–142–43–15–57–144–137–0
	1	Y	0–***152D***–*58*–***152P***–0
	1	C	0–**53D**–180–198–110–4–155–25–55–165–130–54–179–149–0
	2	Y	0–138–154–12–***80P***–150–***80D***–134–24–163–177–109–195–26–105–**53P**–0
	2	C	0–**111P**–50–102–157–33–81–164–34–78–169–129–79–**196P**–0
	1	C	0–**111D**–9–161–71–65–136–35–135–120–185–0
CMT10Y	2	**O**	0–166–**199P**–**125D**–45–**125P**–**199D**–18–0
	2	Y	0–147–5–84–173–17–113–61–93–***104D***–99–***104P***–**96D**–6–0
	1	C	0–183–**96P**–59–85–16–86–141–191–193–91–98–92–151–0
	1	C	0–**53P**–2–115–178–57–15–43–142–42–172–144–137–0
	1	C	0–**53D**–180–21–73–72–171–74–75–133–22–41–145–40–0
	1	C	0–105–149–179–110–4–155–25–55–165–130–54–109–**154D**–0
	2	Y	0–26–195–134–24–163–150–80–177–12–***138D***–***154P***–***138P***–0
	1	C	0–196–77–3–158–79–129–169–29–121–68–116–184–**28D**–0
	2	C	0–**111D**–50–102–157–33–81–120–164–34–78–185–76–**28P**–0
	1	C	0–**111P**–9–135–35–136–65–71–161–103–51–0
	1	C	0–27–176–1–122–128–66–188–20–30–**69D**–132–0
	1	C	0–167–108–126–63–90–32–131–160–70–101–**69P**–0
CMT13X	1	C	0–**99D**–98–59–65–61–57–54–52–110–97–95–0
	1	C	0–**99P**–40–43–45–48–51–50–49–44–41–37–0
	1	C	0–109–26–32–35–36–34–31–28–23–20–**89D**–0
	1	C	0–**89P**–114–17–22–24–27–33–30–25–19–16–0
CMT13Y	1	C	0–**105D**–106–103–73–76–77–78–80–79–68–98–99–0
	2	C	0–**94D**–**97D**–115–40–43–45–59–57–54–52–53–0
	3	C	0–**94P**–41–44–46–49–47–50–51–48–42–**39P**–**97P**–0
	5	C	0–**105P**–107–104–116–110–**39D**–38–37–109–114–**90D**–**91D**–**87P**–0
	1	C	0–**92D**–26–28–31–30–33–34–36–35–32–29–0
	1	C	0–**87D**–17–16–19–25–22–24–27–23–20–21–0
	3	C	0–86–85–84–5–4–3–6–118–18–**90P**–**91P**–**92P**–0
CMT14X	1	C	0–**43D**–42–44–45–46–48–51–50–52–49–47–0
	1	C	0–67–65–66–62–74–72–61–64–68–41–**43P**–0

Notes. *t*: number of split customers on the route, S: route shape, C: cycle, Y: cycle with co-located customer pairs, L: lasso, O: other, D: delivery service only, P: pickup service only. Split customers are shown in bold; co-located customers are shown in italics.

instances 4, 5, 9 and 10, plays a part here. For example, on a cursory look at the first route in CMT5X, it appears that the route zigzags between customers 10 and 189. A closer look reveals that these customers are located at the same coordinates. Hence, if we represent both with a single vertex, this route actually will have a cyclical shape. To highlight this issue, co-located customers are marked in italics in Table 6. Routes that become cyclical, once visiting such customer pairs is considered as a single stop, are denoted by "Y." Including such routes, all but two routes can be described as cyclical shaped. This is a marked difference to studies on the TSPDDP, where lasso and figure-of-eight solutions are common. Of course, in the TSPDDP, a customer cannot be split between two routes, whereas in our experiments, if we disregard co-located customers, only three customers are served by the same vehicle for delivery and pickup, whereas the remaining 46 are split between routes.

One of the remaining routes (the first route in CMT10X) is *lasso shaped*, with one split customer (28) that is served for delivery at the very beginning of the route and for pickup at the very end. Between these stops, there is one delivery-only customer and nine nonsplit customers. The other route (the first route in CMT10Y, 0–166–199P–125D–45–125P–199D–18–0, length 58) has a more surprising shape and even has a pickup before a delivery. On closer observation, we notice that all its customers are placed nearly on a straight line. Due to using integer distances, this tour has the same length as the optimal (Hamiltonian) TSP tour 0–166–199–125–45–18–0. The total of delivery and pickup demand is much less than the vehicle capacity; thus, the order of deliveries and pickups does not matter.

Split customers tend to occur at the beginning or the end of the routes—which makes sense as they also tend to be near the depot. However, for about a quarter of the routes (19 of 68, 28%), they occur midroute.

Future research on an improved solution algorithm can benefit from these observations. Because customers tend to be split between routes rather than within a route, we should develop an operator that can achieve this. For example, "splitshift" would duplicate a simultaneous customer and insert either its linehaul or backhaul into another route. (In this case, customers would not be duplicated at the start but by this operator.) Such an operator may work best in an environment where infeasible solutions are allowed, because it could help to achieve/restore feasibility. Finally, we must allow split customers to occur freely—allowing them to be placed only at the beginning or the end of a route would be too restrictive.

7. An Improved Solution Method and an Analysis of Its Results

In this section, we utilise the observations of the previous section to create a VRPDDP-specific methodology, and compare it against the method of solving the VRPDDP as a VRPMDP with twice as many customers.

7.1. An Improved Solution Method

Our method will be based on the following observations:

—Doubling the number of customers slows down any heuristic; it may be better to reduce the instance size for only allowing splitting for some of the customers.

—Most split customers occur near the depot.

—Many split customers have a high demand or a high pickup.

—In the data set of our study, there are co-located customers that were often served twice.

—Although many split customers are in a dense cluster, this is a harder-to-identify aspect. Looking back at the data analysed in §6.4 (Table 5), we can see that 28 out of 30 split customers that were in a cluster were already candidates for splitting for one of the above reasons.

—Routes do not follow some particular shape such as lasso, and split customers may occur in any part of the route (although they often occur at the beginning or the end).

—Most split customers are served by a different route for delivery and for pickup.

These observations suggest a solution method where, rather than duplicating all customers, we duplicate only customers that are near the depot (defined as the nearest 25% of customers) and the co-located customers. Customers with high demand or pickup may also be considered, but we need not consider customers in dense clusters. If during the run of the algorithm we wish to allow further customers to be split, it is sensible to allow a splitting operator that would move either the delivery or the pickup of the customer to another route. (We note that merge operators often accompany split operators; however, it may be simpler to omit this.)

The backbone of the improved solution method will still be the tried-and-tested reactive tabu search algorithm of Wassan, Wassan, and Nagy (2008). The main steps of the improved RTS-VRPDDP algorithm are shown as follows.

Step 1: Initialisation phase
Step 2: Neighbourhood search phase
Step 3: RTS updating phase
Step 4: Fine-tuning phase

These steps are explained in detail next. Note that Steps 2–4 are repeated for a fixed number of iterations.

The *initialisation phase* begins with duplicating some of the customers. This means that for each customer that lies near the depot or is co-located with another customer, we create two fictitious customers: one with zero pickup and the other with zero delivery. (However, if a near-depot or co-located customer happens to have zero delivery or pickup, then it will not be duplicated.) Optionally, we may also duplicate customers with high demand or pickup (defined as being in the top 25% when sorted according to either of these values). We have two ways of obtaining an initial solution. Firstly, we may simply take the corresponding VRPSDP solution as our initial solution, as was done in §6. In the VRPDDP solution, each customer that is duplicated will have its delivery entity followed by its pickup entity. Second, we can use the modified sweep method (see Wassan, Wassan, and Nagy 2008 for details) to obtain a feasible solution to the VRPDDP. Finally, reactive tabu search parameters are initialised for parameter settings (see Wassan, Wassan, and Nagy 2008).

The *neighbourhood search phase* of the algorithm (Step 2) is built around two well-known moves, namely *shift* and *swap* (for more details, see Wassan, Wassan, and Nagy 2008) and an optional new operator called *splitshift*. *Splitshift* considers only customers that are currently served in a single visit, splits them into a delivery and a pickup entity, and inserts either the delivery or the pickup entity (whichever gives a better solution) to the best possible position on another route. We note that no merge operator is used; if a delivery and pickup entity of a customer should find themselves next to each other in subsequent moves, this is fine by us, but we do not think that a separate operator is required to bind them together again. In Step 2, the entire neighbourhood defined by the moves shift and swap is evaluated. If the best move found is not tabu (see tabu definition later), or is tabu but surpasses our aspiration criterion (i.e., it yields a better solution than the best one recorded), it is carried out. Otherwise, the best nontabu move is implemented. Note that the tabu search framework allows for nonimproving moves.

In the *reactive tabu search updating phase*, we define the tabu status of moves using a tabu list. If in Step 2 customer i was removed from route r (no matter by which of the three operators), we put (i, r) onto the tabu list for the next *tls* iterations, where *tls* is the size of the tabu list (also known as *tabu tenure*). This means that customer i cannot re-enter route r unless the aspiration criterion is met. If a splitshift operator is applied, then either the delivery or the pickup entity of the split customer (whichever was moved to a new route) is placed on the tabu list; however, all of its previous occurrences are removed from the tabu list. The motivation for this is that the two customers are different from their previous nonsplit incarnation and should be allowed to be placed freely in subsequent moves. The remainder of this phase is concerned with dynamically updating the value of *tls*; this aspect is carried out exactly the same way as in Wassan, Wassan, and Nagy (2008), where the reader is referred for further details. Moreover, the parameter settings used in this research are the same as given in that paper.

In the *fine-tuning phase* two operators, *reverse* and *local-shift* (see Wassan, Wassan, and Nagy 2008 for details), are applied, in turn, repeatedly to the two routes that were affected in Step 2, until no improvement is found. Note that tabu status of customers is neither checked nor updated during this phase.

The different versions of our algorithm are created as follows:

—Customers with high demand or supply are duplicated (D) or not (N).

—The initial solution is taken from the VRPSDP solution (V) or obtained by modified sweep (S).

—Operator splitshift is applied (A) or omitted (O).

This gives rise to eight versions, namely, DVA, DVO, DSA, DSO, NVA, NVO, NSA, and NSO.

7.2. Computational Experiments

We again used the Salhi and Nagy (1999) data set and its variants, as defined in §6.2. All instances in Set 1 and Set 2 were solved using all eight versions of our new solution algorithm. We did not think that this algorithm was appropriate to solve instances where in the VRPDDP solution the vast majority of customers would be split, such as those of Set 3. Our experiments confirmed this hypothesis; on Set 3 the new algorithm never improves on the original one. Therefore, our analysis will be restricted to the first two data sets. The algorithms were coded in Fortran 90 and run on the same computer as detailed in §6.2, again for 1,500 iterations. We report on $2 \times 8 \times 28 = 448$ experiments in total.

The results for the two sets are reported in Tables 7 and 8, respectively. Table 9 gives the computing times for ease of reference, including computing times for the VRP and VRPSDP. (We note that because we already have the solution to the VRPDDP, versions that use this as a starting solution do not recreate this, but read it in from a data file. Hence, the solution time of VRPSDP is not included in the computing times of these versions.) In these tables and henceforth, we refer to the methodology described in §6.1 of duplicating all customers and solving the VRPDDP as a VRPMDP as the "original" method. For each algorithm version, we present the total route length z,

Table 7 **Results of the Improved Solution Method (Set 1)**

	Original		DVA			DVO			DSA			DSO			NVA			NVO			NSA			NSO			Best impr.	
Instances	z	k	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	In z (%)	In Δz (%)
CMT1X	478	3	478	3	0.00	478	3	0.00	478	3	0.00	488	3	−2.09	478	3	0.00	478	3	0.00	478	3	0.00	478	3	0.00	0.00	0.00
CMT1Y	476	3	476	3	0.00	476	3	0.00	476	3	0.00	476	3	0.00	476	3	0.00	476	3	0.00	476	3	0.00	476	3	0.00	0.00	—
CMT2X	712	7	712	7	0.00	712	7	0.00	712	7	0.00	713	7	−0.14	712	7	0.00	712	7	0.00	712	7	0.00	712	7	0.00	0.00	0.00
CMT2Y	694	7	694	7	0.00	694	7	0.00	694	7	0.00	694	7	0.00	694	7	0.00	694	7	0.00	694	7	0.00	694	7	0.00	0.00	0.00
CMT3X	726	5	726	5	0.00	726	5	0.00	726	5	0.00	726	5	0.00	726	5	0.00	727	5	−0.14	727	5	−0.14	727	5	−0.14	0.00	—
CMT3Y	723	5	723	5	0.00	723	5	0.00	723	5	0.00	723	5	0.00	723	5	0.00	723	5	0.00	723	5	0.00	729	5	−0.83	0.00	—
CMT4X	901	8	901	8	0.00	901	8	0.00	901	8	0.00	901	8	0.00	901	8	0.00	901	8	0.00	901	8	0.00	901	8	0.00	0.00	0.00
CMT4Y	859	7	859	7	0.00	859	7	0.00	859	7	0.00	876	7	−1.98	859	7	0.00	859	7	0.00	859	7	0.00	875	8	−1.86	0.00	0.00
CMT5X	1,083	11	**1,080**	11	0.28	1,090	11	−0.65	**1,082**	11	0.09	1,090	11	−0.65	**1,081**	11	0.18	1,090	11	−0.65	**1,081**	11	0.18	1,111	11	−2.59	**0.28**	**8.57**
CMT5Y	1,052	10	1,053	10	−0.10	1,053	10	−0.10	1,052	10	0.00	1,053	10	−0.10	1,053	10	−0.10	1,053	10	−0.10	1,052	10	0.00	1,053	10	−0.10	0.00	0.00
CMT6X	555	6	555	6	0.00	555	6	0.00	555	6	0.00	555	6	0.00	555	6	0.00	555	6	0.00	555	6	0.00	555	6	0.00	0.00	—
CMT6Y	556	6	556	6	0.00	556	6	0.00	556	6	0.00	556	6	0.00	556	6	0.00	556	6	0.00	556	6	0.00	556	6	0.00	0.00	—
CMT7X	899	11	899	11	0.00	899	11	0.00	904	11	−0.56	904	11	−0.56	899	11	0.00	899	11	0.00	899	11	0.00	899	11	0.00	0.00	0.00
CMT7Y	902	11	**898**	11	0.44	902	11	0.00	**898**	11	0.44	**900**	11	0.22	**898**	11	0.44	902	11	0.00	**901**	11	0.11	929	12	−2.99	**0.44**	**100.00**
CMT8X	874	9	874	9	0.00	874	9	0.00	885	9	−1.26	929	10	−6.29	874	9	0.00	874	9	0.00	929	10	−6.29	911	10	−4.23	0.00	0.00
CMT8Y	867	9	867	9	0.00	867	9	0.00	867	9	0.00	867	9	0.00	867	9	0.00	867	9	0.00	867	9	0.00	867	9	0.00	0.00	—
CMT9X	1,193	15	**1,171**	**14**	1.84	1,193	15	0.00	**1,172**	15	1.76	**1,192**	15	0.08	**1,171**	**14**	1.84	1,193	15	0.00	**1,172**	15	1.76	1,193	15	0.00	**1.84**	**100.00**
CMT9Y	1,215	15	1,215	15	0.00	1,215	15	0.00	1,216	15	−0.08	1,216	15	−0.08	1,215	15	0.00	1,215	15	0.00	1,215	15	0.00	1,217	15	−0.16	0.00	—
CMT10X	1,438	19	**1,429**	**18**	0.63	1,438	19	0.00	**1,429**	**18**	0.63	1,438	19	0.00	**1,429**	**18**	0.63	1,438	19	0.00	**1,429**	**18**	0.63	1,438	19	0.00	**0.63**	**50.00**
CMT10Y	1,452	19	**1,449**	19	0.21	**1,449**	19	0.21	**1,449**	19	0.21	**1,449**	19	0.21	1,467	19	−1.03	1,452	19	0.00	1,467	19	−1.03	1,467	19	−1.03	**0.21**	**7.32**
CMT11X	1,009	5	1,009	5	0.00	1,009	5	0.00	1,009	5	0.00	1,014	6	−0.50	1,009	5	0.00	1,009	5	0.00	1,009	5	0.00	1,014	6	−0.50	0.00	0.00
CMT11Y	905	4	905	4	0.00	905	4	0.00	905	4	0.00	905	4	0.00	905	4	0.00	905	4	0.00	905	4	0.00	905	4	0.00	0.00	—
CMT12X	680	6	680	6	0.00	680	6	0.00	**679**	6	0.15	680	6	0.00	680	6	0.00	680	6	0.00	**677**	6	0.44	680	6	0.00	**0.44**	**6.67**
CMT12Y	632	5	632	5	0.00	632	5	0.00	632	5	0.00	632	5	0.00	632	5	0.00	632	5	0.00	632	5	0.00	636	5	−0.63	0.00	0.00
CMT13X	1,644	11	1,644	11	0.00	1,644	11	0.00	1,783	12	−8.45	1,783	12	−8.45	1,644	11	0.00	1,644	11	0.00	1,783	12	−8.45	1,783	12	−8.45	0.00	—
CMT13Y	1,708	12	**1,701**	**11**	0.41	**1,707**	12	0.06	**1,707**	12	0.06	1,708	12	0.00	**1,698**	**11**	0.59	**1,698**	**11**	0.59	**1,698**	**11**	0.59	**1,702**	**11**	0.35	**0.59**	**83.33**
CMT14X	831	10	831	10	0.00	842	10	−1.32	**823**	10	0.96	**826**	10	0.60	831	10	0.00	842	10	−1.32	**823**	10	0.96	**826**	10	0.60	**0.96**	**100.00**
CMT14Y	854	11	854	11	0.00	854	11	0.00	854	11	0.00	854	11	0.00	854	11	0.00	854	11	0.00	858	11	−0.47	888	12	−3.98	0.00	0.00
Average impr. (%)				**0.13**			**−0.06**			**−0.22**			**−0.70**			**0.09**			**−0.06**			**−0.42**			**−0.95**		**0.19**	**23.99**
Number of impr.				**6**			**2**			**8**			**4**			**5**			**1**			**7**			**2**		**8**	
Number of best				**5**			**1**			**4**			**1**			**4**			**1**			**4**			**0**		—	

Notes. z: total route length, k: number of vehicles, Δz: detour cost, impr. = improvement: $(z(\text{new})-z(\text{original}))/z(\text{original})$, improved solutions highlighted in **bold**.

Table 8 Results of the Improved Solution Method (Set 2)

	Original		DVA			DVO			DSA			DSO			NVA			NVO			NSA			NSO			Best impr.	
Instances	z	k	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	z	k	Impr. (%)	In z (%)	In Δz (%)
CMT1X	1,169	19	**1,160**	19	0.77	**1,166**	19	0.26	**1,166**	19	0.26	**1,166**	19	0.26	**1,166**	19	0.26	1,185	19	−1.37	1,169	19	0.00	1,185	19	−1.37	**0.77**	**8.65**
CMT1Y	1,093	18	1,100	18	−0.64	1,100	18	−0.64	**1,086**	**17**	0.64	1,100	18	−0.64	**1,076**	**17**	1.56	1,099	18	−0.55	1,100	18	−0.64	1,099	18	−0.55	**1.56**	**11.33**
CMT2X	1,978	36	2,007	**35**	−1.47	2,040	**35**	−3.13	2,014	**35**	−1.82	2,040	**35**	−3.13	2,007	**35**	−1.47	2,074	38	−4.85	2,003	36	−1.26	2,143	38	−8.34	0.00	0.00
CMT2Y	1,903	32	**1,895**	32	0.42	1,903	32	0.00	**1,895**	**31**	0.42	1,924	33	−1.10	**1,889**	**31**	0.74	1,980	33	−4.05	**1,886**	32	0.89	1,980	33	−4.05	**0.89**	**6.23**
CMT3X	2,009	30	**1,992**	**29**	0.85	2,009	30	0.00	**1,996**	**29**	0.65	2,009	30	0.00	**1,993**	**29**	0.80	**1,990**	**29**	0.95	**2,002**	**29**	0.35	**1,990**	**29**	0.95	**0.95**	**6.55**
CMT3Y	1,860	28	**1,859**	**26**	0.05	**1,841**	**27**	1.02	**1,835**	**27**	1.34	**1,841**	**27**	1.02	**1,846**	**27**	0.75	1,873	**27**	−0.70	1,861	28	−0.05	1,873	**27**	−0.70	**1.34**	**14.20**
CMT4X	2,806	45	**2,802**	45	0.14	**2,799**	**44**	0.25	**2,792**	45	0.50	**2,799**	**44**	0.25	**2,781**	**44**	0.89	2,870	46	−2.28	2,808	45	−0.07	2,870	46	−2.28	**0.89**	**6.04**
CMT4Y	2,605	42	**2,584**	**41**	0.81	2,606	42	−0.04	**2,601**	42	0.15	2,606	42	−0.04	**2,599**	42	0.23	2,612	42	−0.27	2,605	42	0.00	2,690	42	−3.26	**0.81**	**6.69**
CMT5X	3,843	68	**3,691**	**62**	3.96	**3,736**	**63**	2.78	**3,687**	**63**	4.06	**3,736**	**63**	2.78	**3,679**	**62**	4.27	3,865	68	−0.57	**3,681**	**62**	4.22	4,558	**66**	−18.61	**4.27**	**32.48**
CMT5Y	3,371	58	**3,359**	**57**	0.36	3,403	59	−0.95	**3,367**	**57**	0.12	3,372	**57**	−0.03	**3,365**	**57**	0.18	3,403	59	−0.95	**3,353**	**57**	0.53	4,329	**57**	−28.42	**0.53**	**5.01**
CMT6X	1,169	19	**1,160**	19	0.77	**1,166**	19	0.26	**1,166**	19	0.26	**1,166**	19	0.26	**1,166**	19	0.26	1,185	19	−1.37	1,169	19	0.00	1,185	19	−1.37	**0.77**	**8.65**
CMT6Y	1,093	18	1,100	18	−0.64	1,100	18	−0.64	**1,086**	**17**	0.64	1,100	18	−0.64	**1,076**	**17**	1.56	1,099	18	−0.55	1,100	18	−0.64	1,099	18	−0.55	**1.56**	**11.04**
CMT7X	2,027	36	**1,995**	**35**	1.58	2,027	36	0.00	**2,016**	**35**	0.54	2,057	36	−1.48	**2,013**	**35**	0.69	2,083	38	−2.76	**1,990**	**35**	1.83	2,130	37	−5.08	**1.83**	**16.67**
CMT7Y	1,903	32	**1,895**	32	0.42	1,903	32	0.00	**1,895**	**31**	0.42	1,924	33	−1.10	**1,889**	**31**	0.74	1,980	33	−4.05	**1,886**	32	0.89	1,980	33	−4.05	**0.89**	**6.23**
CMT8X	2,009	30	**1,992**	**29**	0.85	2,009	30	0.00	**1,996**	**29**	0.65	2,009	30	0.00	**1,993**	29	0.80	**1,990**	**29**	0.95	**2,002**	**29**	0.35	**1,990**	**29**	0.95	**0.95**	**6.55**
CMT8Y	1,860	28	**1,859**	**26**	0.05	**1,841**	**27**	1.02	**1,835**	**27**	1.34	**1,841**	**27**	1.02	**1,846**	**27**	0.75	1,873	27	−0.70	1,861	28	−0.05	1,873	27	−0.70	**1.34**	**14.20**
CMT9X	2,806	45	**2,802**	45	0.14	**2,799**	**44**	0.25	**2,792**	45	0.50	**2,799**	**44**	0.25	**2,781**	**44**	0.89	2,870	46	−2.28	2,808	45	−0.07	2,870	46	−2.28	**0.89**	**6.13**
CMT9Y	2,605	42	**2,584**	**41**	0.81	2,605	42	0.00	**2,601**	42	0.15	2,606	42	−0.04	**2,599**	42	0.23	2,612	42	−0.27	2,605	42	0.00	2,690	42	−3.26	**0.81**	**6.67**
CMT10X	3,843	68	**3,691**	**62**	3.96	**3,736**	**63**	2.78	**3,687**	**63**	4.06	**3,736**	**63**	2.78	**3,679**	**62**	4.27	3,865	68	−0.57	**3,681**	**62**	4.22	4,558	**66**	−18.61	**4.27**	**32.48**
CMT10Y	3,371	58	**3,359**	**57**	0.36	3,371	58	0.00	**3,367**	**57**	0.12	3,372	**57**	−0.03	**3,365**	**57**	0.18	3,403	59	−0.95	**3,353**	**57**	0.53	4,329	**57**	−28.42	**0.53**	**5.01**
CMT11X	3,894	31	**3,856**	**29**	0.98	**3,866**	**29**	0.72	**3,867**	**29**	0.69	**3,866**	**29**	0.72	**3,865**	**29**	0.74	3,941	32	−1.21	**3,862**	**29**	0.82	4,005	**30**	−2.85	**0.98**	**4.39**
CMT11Y	3,309	28	3,313	28	−0.12	3,309	28	0.00	3,323	29	−0.42	3,323	29	−0.42	3,324	29	−0.45	3,333	29	−0.73	3,313	28	−0.12	3,544	28	−7.10	0.00	0.00
CMT12X	2,535	34	**2,475**	**32**	2.37	2,534	35	0.04	**2,506**	**33**	1.14	2,534	35	0.04	**2,517**	**33**	0.71	2,583	35	−1.89	**2,518**	**33**	0.67	2,583	35	−1.89	**2.37**	**9.30**
CMT12Y	2,233	32	2,248	32	−0.67	2,265	33	−1.43	2,233	32	0.00	2,265	33	−1.43	2,238	32	−0.22	2,250	32	−0.76	2,256	32	−1.03	2,250	32	−0.76	0.00	0.00
CMT13X	3,894	31	**3,856**	**29**	0.98	**3,866**	**29**	0.72	**3,867**	**29**	0.69	**3,866**	**29**	0.72	**3,865**	**29**	0.74	3,941	32	−1.21	**3,862**	**29**	0.82	4,005	**30**	−2.85	**0.98**	**4.39**
CMT13Y	3,309	28	3,313	28	−0.12	3,309	28	0.00	3,323	29	−0.42	3,323	29	−0.42	3,324	29	−0.45	3,333	29	−0.73	3,313	28	−0.12	3,544	28	−7.10	0.00	0.00
CMT14X	2,535	34	**2,475**	**32**	2.37	**2,534**	35	0.04	**2,506**	**33**	1.14	**2,534**	35	0.04	**2,517**	**33**	0.71	2,583	35	−1.89	**2,518**	**33**	0.67	2,583	35	−1.89	**2.37**	**9.30**
CMT14Y	2,233	32	2,248	32	−0.67	2,265	33	−1.43	2,233	32	0.00	2,265	33	−1.43	2,238	32	−0.22	2,250	32	−0.76	2,256	32	−1.03	2,250	32	−0.76	0.00	0.00
Average impr. (%)				**0.67**			**0.07**			**0.64**			**−0.06**			**0.72**			**−1.30**			**0.42**			**−5.54**		**1.16**	**8.51**
Number of impr.				**21**			**11**			**23**			**11**			**23**			**2**			**13**			**2**		**23**	
Number of best				**8**			**0**			**2**			**0**			**6**			**2**			**5**			**2**		—	

Notes. z: total route length, k: number of vehicles, Δz: detour cost, impr. = improvement: $(z(\text{new}) - z(\text{original}))/z(\text{original})$, improved solutions highlighted in **bold**.

Table 9 Computing Times (in Seconds)

Instances	VRP	VRPSDP	Original	DVA	DVO	DSA	DSO	NVA	NVO	NSA	NSO
Set 1											
CMT1X	28	32	34	33	31	35	34	31	31	32	33
CMT1Y	29	31	40	38	37	40	36	36	36	38	38
CMT2X	36	42	44	42	41	42	43	37	39	42	40
CMT2Y	33	38	51	47	47	49	46	47	46	47	49
CMT3X	163	174	142	133	140	139	135	130	127	131	132
CMT3Y	145	143	133	124	131	136	129	124	122	127	126
CMT4X	247	256	177	174	160	183	166	151	156	164	167
CMT4Y	182	264	172	156	167	180	165	157	149	170	159
CMT5X	306	257	340	320	336	352	336	310	305	315	339
CMT5Y	284	266	351	328	340	361	319	309	320	328	331
CMT6X	20	20	26	25	25	25	23	24	24	24	24
CMT6Y	21	25	26	25	24	26	24	22	24	25	25
CMT7X	24	21	50	46	45	50	48	46	43	49	45
CMT7Y	25	26	52	46	47	51	50	47	47	47	49
CMT8X	63	63	125	121	119	129	122	107	111	116	124
CMT8Y	66	68	128	123	127	131	126	119	119	117	116
CMT9X	92	93	186	173	173	183	179	169	164	174	181
CMT9Y	90	99	187	171	185	183	170	173	164	168	169
CMT10X	138	139	209	208	192	204	189	184	189	201	188
CMT10Y	127	150	201	191	186	202	181	188	186	183	194
CMT11X	43	215	161	154	149	153	151	138	147	156	156
CMT11Y	48	200	162	154	148	165	160	140	142	153	148
CMT12X	57	99	74	71	70	73	67	69	64	66	68
CMT12Y	63	102	78	73	75	77	74	70	72	77	72
CMT13X	45	65	134	131	132	128	130	116	114	125	131
CMT13Y	44	67	131	120	130	137	124	118	117	121	120
CMT14X	52	77	75	72	70	75	73	66	64	68	70
CMT14Y	52	75	80	72	75	77	74	71	74	73	72
Average	90	111	127	120	122	128	121	114	114	119	120
Set 2											
CMT1X	5	26	29	26	25	30	28	24	25	26	28
CMT1Y	5	27	33	33	31	33	31	29	28	33	33
CMT2X	11	35	36	35	33	36	37	32	32	34	34
CMT2Y	11	32	41	41	39	41	38	41	39	40	42
CMT3X	27	156	114	115	119	112	116	113	104	116	112
CMT3Y	35	121	112	104	111	121	104	110	100	103	107
CMT4X	25	215	151	149	130	152	136	129	136	137	147
CMT4Y	26	220	140	127	133	144	138	139	126	142	131
CMT5X	39	220	285	279	295	295	300	277	261	275	303
CMT5Y	45	230	283	289	276	316	260	265	279	270	288
CMT6X	5	17	21	20	21	21	19	20	21	21	20
CMT6Y	5	21	22	21	20	21	19	19	19	21	20
CMT7X	10	17	44	39	38	42	39	40	35	39	38
CMT7Y	10	22	43	38	41	45	42	38	41	39	43
CMT8X	26	55	111	99	97	104	102	91	94	100	109
CMT8Y	21	60	113	102	102	108	101	98	98	104	94
CMT9X	48	77	149	153	153	157	153	150	140	156	149
CMT9Y	45	79	162	150	163	152	144	148	147	136	149
CMT10X	49	112	180	176	172	164	160	157	155	168	168
CMT10Y	47	130	174	157	151	164	151	160	160	153	155
CMT11X	27	179	144	132	129	127	125	122	127	137	127
CMT11Y	35	174	145	136	126	140	128	116	126	132	132
CMT12X	35	85	60	63	61	60	55	57	52	53	58
CMT12Y	38	90	65	61	63	64	59	58	58	66	60
CMT13X	35	52	108	107	117	112	108	102	93	112	113
CMT13Y	30	54	106	104	104	116	105	95	100	104	105
CMT14X	39	61	60	63	58	65	63	56	55	58	59
CMT14Y	45	66	65	61	64	63	61	61	63	59	60
Average	28	94	107	103	103	107	101	98	97	101	103

the number of vehicles k, and the improvement from the original method defined as $(z(\text{original}) - z(\text{new}))/z(\text{original})$. To ease comparison, the results of the original method are also displayed in the tables. For each instance, the best improvement achieved is tabulated in the penultimate column. Whereas including detour lengths and improvements in Δz for each algorithm version would have made the tables very cumbersome, we decided to include the best improvement in detour cost in the last column. This is defined as $(\Delta z(\text{original}) - \Delta z(\text{best}))/\Delta z(\text{original})$. (As $\Delta z(\text{original}) - \Delta z(\text{new}) = z(\text{original}) - z(\text{new})$, the improvement in detour cost for each instance is just a constant multiple of the improvement in route length.) Our analysis will be based to a large extent on three summary measures: the average of the percentage improvements, the number of instances for which an improvement on the original method is obtained, and the number of instances for which the best results have been obtained. However, we only count a result as "best" if it is strictly better than the result of the original method—otherwise all versions would appear "best" if no improvement was achieved or even achievable, distorting the comparison of algorithm versions. These measures are displayed for each algorithm version in the last three rows of Tables 7 and 8.

In the next two sections we present an analysis of these results. Section 7.3 compares the new algorithm versions to each other to identify which aspects make the algorithm more successful. Section 7.4 asks whether the new algorithms are more efficient than the methodology presented and analysed in §6.

7.3. An Analysis of the Different Algorithm Versions

From the testing of the different versions we can derive indications on which ones are best. Within the eight versions of our algorithm, there are large variations. In Set 1, only DVA and NVA give positive average improvements on the results of the original algorithm (0.13% and 0.09%, respectively). In terms of both the number of instances on which an improvement is found and the number of instances for which the best result is found, the versions where split-shift is applied (DVA, DSA, NVA, and NSA) appear the to be best. For the eight instances on which an improvement is obtained, at least one of these versions arrives at the best result. Taking a closer look, we can see that the reason DSA and NSA do not give positive improvements on average is that their average result is pulled down by the poor results on instance CMT13X (and also CMT8X in the case of NSA). In Set 2, five versions give positive average improvements, with the approximately equal front-runners being DVA, DSA, and NVA (with 0.67%, 0.64%, and 0.72%, respectively). For 16 of the 23 instances on which an improvement is obtained, at least one of these versions arrives at the best result. The supremacy of these versions is also confirmed by both the number of instances for which an improvement is found and (except for DSA) the number of instances for which the best result is obtained. NSA is clearly in fourth place, with a 0.42% average improvement, also according to the number of improved solutions found. However, it often yields better results than DVA, DSA, and NVA; it obtains the best results for CMT2Y, CMT5Y, CMT7X, CMT7Y, and CMT10Y. (We note that the best results for CMT3X and CMT8X are obtained by otherwise poorly performing NVO and NSO.)

If we had to work with just one version, it would be DVA. It has the best average improvement on Set 1 and comes a close second-best on Set 2. Out of the 31 instances on the two data sets where a positive improvement is obtained, DVA finds an improvement 27 times, is best 13 times, and lags behind the best result on the other 18 instances by 0.69% on average. At the other end of the spectrum, we observe NSO, which has several very poor results. In Set 2, for nine instances it finds solutions more than 5% worse than the original method, including two solutions (CMT5Y and its capacitated counterpart CMT10Y) that are nearly 30% worse. For most instances, NSO gives worse results than the original method, except for CMT13Y and CMT14X in Set 1 and CMT3X and CMT8X in Set 2 (interestingly, for the latter two NSO finds the best result).

We can explain these variations by looking at the three aspects in which the algorithm versions differ. First, let us look at whether it is a good idea to duplicate at the outset the customers with high delivery or pickup demand. The average improvement of the versions that do so over both data sets is 0.06%, whereas for those that do not do so is −0.88%. (The latter result may, however, be partly due to the very bad results found by NSO.) The benefit of this idea is also shown by the fact that DVA, DVO, DSA, and DSO improve on the original method 86 times (out of $4 \times 28 = 112$ experiments), whereas the other four do so only 55 times. However, when looking at the best solutions found, the situation changes. The versions where duplication of high delivery/pickup demand customers occur find the best solution only 21 times, whereas the others do so 24 times. This shows that reducing the number of customers by duplicating fewer customers at the outset may speed the algorithm toward good-quality solutions; however, this should be offset by allowing splitting of customers during the run of the algorithm.

Second, let us look at the initial solution—which is better, starting from the VRPSDP solution or creating one using modified sweep? The average improvement over both data sets of the versions that use VRPSDP is 0.04%, whereas for those that use modified sweep it is −0.82%. Looking at the best results obtained, the VRPSDP-based solutions appear better (27 best results) than the sweep-based ones (18 solutions). However, if we look at the number of instances where improvements are obtained, the two groups of versions appear equally good (72 and 71 improvements, respectively). Therefore, although using the VRPSDP solution as the initial solution is preferred, using sweep allows the search to traverse different solutions and is as likely to find a relatively good-quality solution as using VRPSDP. It seems that an added advantage of using modified sweep is that the VRPSDP solution need not be calculated. However, in practice one would always be interested in finding this solution. It also serves as an upper bound, which is useful because sweep-based methods sometimes (35 times out of 224 experiments) find a result that is worse than the VRPSDP solution. (In fact, one should also always solve the corresponding VRP instance; this gives a lower bound and the detour cost. Moreover, if $z(\text{VRP}) = z(\text{VRPSDP})$, then there is no need to solve the VRPDDP, because no splitting is needed.)

Finally, let us look at the benefits of the splitshift operator. Here, at last, the situation is clear. Versions that use this operator give an average improvement of 0.36% over both data sets, whereas the others yield an improvement of −1.08%. Splitshift-based versions find 80 improved solutions, 38 of which are best solutions; the others only find 26 improvements, seven of which are best results. Therefore, we can definitely recommend the use of the splitshift operator. The reason for its success is that it allows customers to be split that are not near the depot or do not have a high delivery or pickup demand; but does not unnecessarily duplicate such customers.

Although the focus of this analysis was on route length, because the algorithm is geared toward minimising this, a few comments on the number of vehicles are appropriate. Our experiments show little variation in the number of vehicles found by different versions of the solution algorithm. In Set 1, k differed by no more than 1 from that found by the original algorithm. In Set 2, with two exceptions, k differed by no more than 2 from that found by the original algorithm. (The exceptions were CMT5X and its capacitated counterpart CMT10X, for which the original algorithm found a rather poor solution.) Normally, changes in k followed changes in z, but there were several examples in Set 2 (CMT2X, CMT3Y, CMT5Y, CMT8Y, CMT10Y, CMT11X, and CMT13X), where z increased yet k decreased.

Finally, we can see from Table 9 that there is very little difference in computing times among the different versions—each take an average of about two minutes per instance. Versions NVA and NVO appear to be quickest, presumably due to having fewer customers duplicated and not requiring to use modified sweep for an initial solution.

7.4. Is the New Method Really an "Improved" One?

Let us now compare the original method of duplicating all customers with the new algorithm. We have already seen that some versions improve on the original, on average. Looking at Tables 7 and 8, we see that for every instance, except CMT2X of Set 2, a new best solution was found. (On this instance even the best result, found by NSA, is 1.26% worse than that of the original method.) Moreover, we see that no version is always as good as the original method, although DVA is never more than 0.10% worse on Set 1 and (if we discount CMT2X) never more than 0.67% worse on Set 2. Likewise, no version is always worse than the original one (although NVO only improves on the original method for three instances out of 56). Looking in more detail at the "best version" DVA, we observe that it improves on the original method by 0.13% on Set 1 and 0.67% on Set 2, and finds improved solutions for six instances (out of a possible 19) on Set 1 and 21 instances on Set 2.

Comparing the results of the two data sets, we notice that the improvements are much larger on Set 2. This is due to the detour costs in Set 1 often being small, so there was only a small range for improvement. Moreover, in Set 1, the measure of "number of improved solutions" should be seen in the light of the detour cost being zero for nine instances; hence, improvements were only achievable on the remaining 11 instances.

So far, we have looked only at improvements in z. Previously, we have stressed the importance of the detour cost Δz as a better indicator. However, when comparing different algorithm versions it is simpler to focus only on z, because the two measures are just constant multiples of each other. Nonetheless, we also included in Tables 7 and 8 some indication of how much these improvements are in terms of Δz, in the form of "best improvement in Δz." Looking at the last columns of these tables, we can see that the improvements achieved by the new algorithm are quite significant in terms of reducing the detour cost—the average improvements on the two data sets are 23.99% and 8.51%, respectively. (These are the averages of the best improvements; the average improvement of the "best version" DVA is 17.01% on Set 1 and 4.11% on Set 2.) In Table 7 one can also easily identify the nine instances with $\Delta z = 0$ where improvements could not have been achieved and the three instances (CMT7Y, CMT9X, and CMT14X) where the new algorithm found a VRPDDP solution with the same cost as the VRP solution, and thus reduced the detour costs by 100%. In Table 8, the best improvement in Δz was achieved on instances CMT5X and CMT10X; the detour cost was reduced by 32.48%.

One may elect to solve the VRPDDP eight times using all the different algorithm versions and select the best solution found. Looking at the penultimate columns of Tables 7 and 8, we can see the improvements that can be achieved by doing so. The average improvement is 0.19% for Set 1 with eight improved solutions and 1.16% for Set 2 with 23 improved solutions, clearly better than applying just one of the eight algorithm versions. (This corresponds to detour cost improvements of 23.99% and 8.51%, respectively.) This, of course, is quite time consuming. Another approach would be to run our algorithm just a few times. For example, the best three-version combination appears to be DVA, DSA, and NVA. This approach would yield the best solution for all instances but one in Set 1. (For CMT12X, it lags behind the best solution found by 0.29%.) For Set 2, it would yield the best solution for 20 instances and lag behind the best solutions by up to 0.25% on the others.

Finally, we can see from Table 9 that the new versions are slightly (about 5%) quicker, on average, than the original method, although still somewhat slower than solving the VRPSDP. Clearly, duplicating about 25% of customers yields a computing time between no duplication (VRPSDP) and duplication of all customers (the original method).

In summary, the new algorithm compares favourably to the methodology of duplicating all customers and solving the resulting VRPMDP, although the improvement is generally small. (However, even a small improvement in z can be considerable in terms of Δz.) Some variation was found among the different algorithm versions, with versions using the operator split-shift being clearly more efficient, and version DVA appearing the best by a narrow margin.

8. Conclusions and Suggestions

We investigated the vehicle routing problem with divisible deliveries and pickups (VRPDDP), a rarely addressed extension of the VRP. We placed the VRPDDP in the context of other VRP extensions and presented both a MILP formulation and a reactive tabu search metaheuristic. Our computational experiments led us to the following four main conclusions.

1. Serving customers twice can often reduce costs and—perhaps even more importantly—the number of vehicles required. It appears that the presence of very small deliveries and pickups is not conducive to splitting. Route length and the number of vehicles are reduced considerably when the delivery and pickup figures vary within a wide range. The benefits of splitting are shown to be even more significant for instances where there is a large difference between delivery and pickup values. Splitting seems more beneficial for clustered instances; however, the presence of a maximum time constraint does not appear to be a predictor for splitting.

2. Three important characteristics of customers who are served twice were observed: they are near the depot, they have a high delivery or pickup demand, or they are located in a dense cluster of customers, with the first factor being especially significant. These observations lead us to believe that good solutions could be achieved if we consider splitting only for customers with such characteristics.

3. Almost all routes take the shape of a cycle, with customers being split across (rather than within) routes. Split customers very often, but not always, occur at the beginning or the end of a route.

4. A promising method for solving the VRPDDP is based on duplicating "promising" customers at the outset and allowing splitting of the remaining customers during the run of the algorithm. The results of such a method compare favourably to the concept of duplicating all customers and solving the VRPDDP as a VRPMDP.

We plan to take this research forward as follows:

1. Extend the scope of our analysis to the split delivery VRPDDP (see Mitra 2005, 2008), allowing customers' delivery and pickup requests to be served in several visits.

2. Merging our lines of research in this paper and in Nagy, Wassan, and Salhi (2013), we wish to investigate the VRPDDP with restricted mixing. This model, introduced by Hoff and Løkketangen (2006), forces customers to be served separately to avoid situations where there is a mixture of delivery and pickup goods on board, but not enough space to have access to both kinds of goods.

Acknowledgments

The authors thank the editor, the associate editor, and the two reviewers, whose helpful and constructive comments have helped to greatly improve the original manuscript. The authors thank the Economic and Social Research Council (ESRC) of the United Kingdom [Grant RES-000-22-3451] for their generous financial assistance toward this project.

References

Anily S (1996) The vehicle routing problem with delivery and back-haul options. *Naval Res. Logist.* 43:415–434.

Archetti C, Speranza MG (2008) The split delivery vehicle routing problem: A survey. Golden B, Raghavan S, Wasil E, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer, New York), 103–122.

Archetti C, Speranza MG (2012) Vehicle routing problems with split deliveries. *Internat. Trans. Oper. Res.* 39:3–22.

Archetti C, Savelsbergh MWP, Speranza MG (2006) Worst-case analysis for split delivery vehicle routing problems. *Transportation Sci.* 40:226–234.

Archetti C, Savelsbergh MWP, Speranza MG (2008) To split or not to split: That is the question. *Transportation Res. Part E* 44: 114–123.

Battiti R, Brunato M, Mascia F (2008) *Reactive Search and Intelligent Optimization* (Springer, Berlin).

Berbeglia G, Cordeau J-F, Gribkovskaia I, Laporte G (2007) Static pickup and delivery problems: A classification scheme and survey. *TOP* 15:1–31.

Christofides N, Eilon S (1969) An algorithm for the vehicle routing dispatching problem. *Oper. Res. Quart.* 20:309–318.

Dror M, Trudeau P (1989) Savings by split delivery routing. *Transportation Sci.* 23:141–145.

Gribkovskaia I, Laporte G, Shyshou A (2008) The single vehicle routing problem with deliveries and selective pickups. *Comput. Oper. Res.* 35:2908–2924.

Gribkovskaia I, Halskau Ø, Laporte G, Vlček M (2007) General solutions to the single vehicle routing problem with pickups and deliveries. *Eur. J. Oper. Res.* 180:568–584.

Halskau Ø, Gribkovskaia I, Myklebost KNB (2001) Models for pickup and deliveries from depots with Lasso solutions. Stefansson G, Tilanus B, eds. *Proc. 13th Annual Conf. Logist. Res. NOFOMA 2001* (Chalmers University of Technology, Göteborg, Sweden), 279–293.

Hoff A, Løkketangen A (2006) Creating Lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search. *Central Eur. J. Oper. Res.* 14:125–140.

Hoff A, Laporte G, Løkketangen A, Gribkovskaia I (2009) Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *Eur. J. Oper. Res.* 192:755–766.

Mitra S (2005) An algorithm for the generalized vehicle routing problem with backhauling. *Asia-Pacific J. Oper. Res.* 22: 153–169.

Mitra S (2008) A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *J. Oper. Res. Soc.* 59:1532–1546.

Mosheiov G (1994) The travelling salesman problem with pick-up and delivery. *Eur. J. Oper. Res.* 79:299–310.

Mosheiov G (1998) Vehicle routing with pickup and delivery: Tour-partitioning heuristics. *Comput. Indust. Engrg.* 34:669–684.

Nagy G, Salhi S (2005) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *Eur. J. Oper. Res.* 162:126–141.

Nagy G, Wassan NA, Salhi S (2013) The vehicle routing problem with restricted mixing of deliveries and pickups. *J. Scheduling* 16:199–213.

Nowak M, Ergun Ö, White CC (2008) Pickup and delivery with split loads. *Transportation Sci.* 42:32–43.

Nowak MA, Ergun Ö, White CC (2009) An empirical study on the benefit of split loads with the pickup and delivery problem. *Eur. J. Oper. Res.* 198:734–740.

Parragh SN, Doerner KF, Hartl RF (2008) A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *J. Betriebswirtschaft* 58:21–51.

Salhi S, Nagy G (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J. Oper. Res. Soc.* 50:1034–1042.

Tarantilis CD, Anagnostopoulou AK, Repoussis PP (2013) Adaptive path relinking for vehicle routing and scheduling problems with product returns. *Transportation Sci.* 47:356–379.

Thangiah SR, Fergany A, Awan S (2007) Real-time split-delivery pickup and delivery time window problems with transfers. *Central Eur. J. Oper. Res.* 15:329–349.

Toth P, Vigo D (2002) *The Vehicle Routing Problem* (Society for Industrial and Applied Mathematics, Philadelphia).

Wassan NA (2007) Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *J. Oper. Res. Soc.* 58:1630–1641.

Wassan NA, Nagy G, Ahmadi S (2008) A heuristic method for the vehicle routing problem with mixed deliveries and pickups. *J. Scheduling* 11:149–161.

Wassan NA, Wassan AH, Nagy G (2008) A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *J. Combin. Optim.* 15:368–386.

Note

This article, "The Vehicle Routing Problem with Divisible Deliveries and Pickups" by Gábor Nagy, Niaz A. Wassan, M. Grazia Speranza, Claudia Archetti (first published in *Articles in Advance*, November 15, 2013, *Transportation Science*, DOI:10.1287/trsc.2013.0501), is now licensed under CC-BY. See the attribution below for more information.