INFORMS TutORials in Operations Research

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

Recent Developments in Modeling and Solving the Split Delivery Vehicle Routing Problem

Damon J. GulczynskiBruce GoldenEdward Wasil

TUTORIALS IN OPERATIONS RESEARCH
INFORMS 2008

 | ISBN 978-1-877640-23-0
DOI 10.1287/educ.1080.0046

Recent Developments in Modeling and Solving the Split Delivery Vehicle Routing Problem

Damon J. Gulczynski
Department of Mathematics, University of Maryland, College Park, Maryland 20742, damon@math.umd.edu

Bruce Golden
Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742, bgolden@rhsmith.umd.edu

Edward Wasil
Kogod School of Business, American University, Washington, D.C., 20016, ewasil@american.edu

Abstract In the split delivery vehicle routing problem, a customer's demand can be split among several vehicles. In the last five years or so, researchers have proposed exact and approximate solution methods, modeled variants with time windows and pickups, and developed large-scale benchmark problems. In this tutorial, we summarize the recent literature on the split delivery vehicle routing problem, describe solution procedures and results of computational experiments, and suggest directions for future research.

Keywords vehicle routing problem; heuristics; mixed-integer program

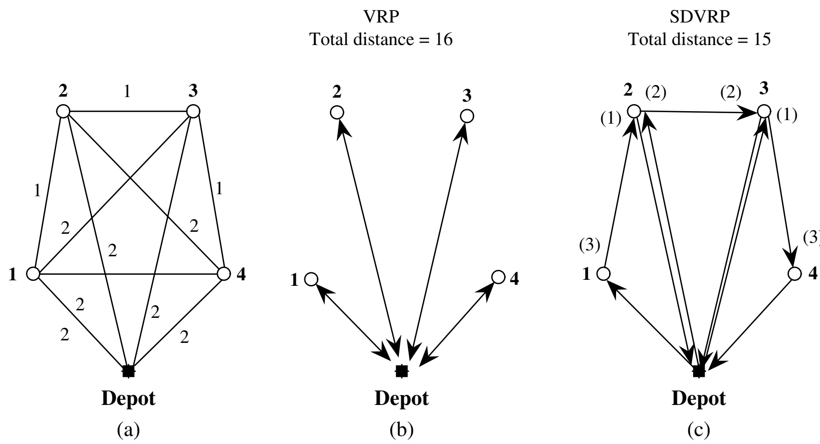
1. Introduction

In the standard version of the vehicle routing problem (VRP), vehicles with the same capacity based at a single depot service many customers. A customer's demand is delivered in one visit by a single vehicle. We must find the minimal cost set of routes for the vehicles that start and end at the depot and do not violate vehicle capacity. The VRP has been studied for nearly 50 years. The book by Golden et al. [15] contains 25 papers that describe the latest applications, algorithms, and computational results.

In the late 1980s, researchers considered the possibility of serving a customer by more than one vehicle to potentially reduce the total distance traveled by the fleet of vehicles. The split delivery vehicle routing problem (SDVRP) retains all features of the standard VRP but allows a customer's demand to be split among several vehicles. In Figure 1, we give an example of the SDVRP with four customers (labeled 1, 2, 3, 4) and a single depot. Each customer has a demand of three units, each vehicle has a capacity of four units, and distances are shown adjacent to edges. In Figure 1(b), the optimal solution to the standard VRP with no split deliveries has one vehicle traveling directly out to each customer, delivering three units, and returning back to the depot for a total distance of 16. In Figure 1(c), split deliveries are allowed. Customers 2 and 3 are now serviced by two different vehicles and the total distance has been reduced to 15.

In the last five years or so, research work on the SDVRP has increased significantly, so that there are currently more than a dozen articles in which the modeling and solving of the SDVRP and its variants (such as the SDVRP with time windows) are addressed. We believe that part of the renewed interest in the SDVRP is due to the increased costs (such as higher fuel and maintenance costs) associated with operating commercial fleets and the need for management to reduce these costs as much as possible. In addition, the availability of powerful metaheuristics has made this problem easier to study computationally. Our goal

FIGURE 1. Splitting deliveries may reduce the distance traveled by a fleet.



Note. Customer demand is three units, vehicle capacity is four units, and edge labels are distances.

in this tutorial is threefold: (1) summarize the open literature on the SDVRP, (2) provide details of solution procedures and report computational results on benchmark problems, and (3) suggest future research directions.

2. Summary of the Recent Literature

The SDVRP was introduced by Dror and Trudeau [11] in 1989. For the next 15 years, there was a steady trickle of published papers, and their algorithmic accomplishments and applications have been described by Chen et al. [7] and Archetti and Speranza [1].

In this section, we summarize recent work on the SDVRP. We focus on the 15 papers given in Table 1 that model and solve the SDVRP and its variants from 2004 to 2008. Our summary of each paper will fall into one of three categories: (1) heuristics, (2) exact methods and bound-generating procedures, and (3) SDVRP variants.

2.1. Heuristics

2.1.1. Tabu Search. Archetti et al. [2] formulate a mixed-integer program (MIP) for the SDVRP in which the quantity delivered on a route cannot exceed a value k (they call this

TABLE 1. Summary of 15 papers that model and solve the SDVRP from 2004 to 2008.

Authors	Year	Algorithm	Variant
Ho and Haugland [16]	2004	Tabu search	Time windows
Mitra [21]	2005	Cheapest-insertion	Backhauls
Archetti et al. [2]	2006	Tabu search	
Lee et al. [19]	2006	Dynamic program, shortest path	Exact algorithm
Boudia et al. [5]	2007	Memetic algorithm	
Chen et al. [7]	2007	MIP, record-to-record travel	
Jin et al. [17]	2007	LP with valid inequalities	Exact algorithm
Mitra [22]	2007	Cluster and route	Backhauls
Mota et al. [23]	2007	Scatter search	
Tavakkoli-Moghaddam et al. [28]	2007	Simulated annealing	Heterogeneous fleet
Thangiah et al. [29]	2007	First insertion, local search	Real-time events
Archetti et al. [3]	2008	IP route optimization	
Jin et al. [18]	2008	Column generation	Bounds generation
Liu et al. [20]	2008	Greedy heuristic, bin-packing	Fixed route
Nowak et al. [24]	2008	Local search, Clarke-Wright	Pickups

problem the k-SDVRP). The authors develop a tabu search algorithm (called SPLITABU) for solving the k-SDVRP. Their algorithm has three phases: (1) initial feasible solution phase—make as many direct trips to customers as possible and then solve a giant traveling salesman tour using the GENIUS algorithm (Gendreau et al. [12]); (2) tabu search phase—remove a customer from a current set of routes and insert it on a new route or an existing route with available capacity in the cheapest way, and consider inserting a customer on a route without removing it from its current route; and (3) final improvement phase—improve the solution from the second phase (apply the GENIUS algorithm to individual routes). SPLITABU has only two parameters that need to be set: the length of the tabu list and the maximum number of iterations. The authors also modify SPLITABU in two ways. Solutions are improved using the node interchanges of Dror and Trudeau [11] and 2-opt (this version is called SPLITABU-DT). The authors limit the run time of the second phase to one minute (this version is called FAST-SPLITABU).

Archetti et al. [2] test their three algorithms on seven problems with 50 to 199 customers. Customer demand in each problem is generated using the rules proposed by Dror and Trudeau [11], and this results in 49 test problems. The authors run each algorithm five times on each problem and compare results to those generated by Dror and Trudeau's [11] algorithm (denoted by DT). Overall, SPLITABU-DT is the best performer. On every problem, it finds a better solution than DT. The best solutions produced by SPLITABU-DT are nearly 5.4% lower on average than the solutions produced by DT.

2.1.2. Genetic Algorithm. Boudia et al. [5] solve the SDVRP using a memetic algorithm with population adjustment (MA|PM) described in Sörensen and Sevaux [27]. The authors create an initial population of VRP solutions with no splits using the Clarke and Wright [10] algorithm and the sweep method of Gillett and Miller [14]. Two parent solutions are selected and offspring are created using crossover. The offspring are converted into a solution to the SDVRP (using a procedure called Split). Solutions are improved using traditional VRP local search procedures including customer exchange and 2-opt moves. The authors also consider moves based on the k-split procedure of Dror and Trudeau [11] to split a customer or change the delivery amounts on each visit. An offspring is selected for improvement with a fixed probability. If an improved offspring is better than the current best solution, it replaces a member of the population. In addition, the authors use a threshold to promote diversity in the population so that an improved solution can enter the population only if it is sufficiently different from existing solutions (the notion of diversity control using a distance measure in the solution space is described in Sörensen and Sevaux [27]).

Boudia et al. [5] apply their MA|PM algorithm to the 49 problems used by Archetti et al. [2] and compare results to SPLITABU-DT. They find that one run of MA|PM improves the SPLITABU-DT solution (average of five runs) in 37 problems and appears to be faster (although the machines are slightly different).

2.1.3. Mixed-Integer Programming with a Routing Metaheuristic. Chen et al. [7] focus on the SDVRP. First, they review applications of the SDVRP and the literature on this topic. Next, they present an innovative solution procedure that combines a mixed-integer program and a routing metaheuristic (namely, the record-to-record travel algorithm). This procedure is referred to as EMIP + VRTR. In computational experiments, EMIP + VRTR clearly outperforms SPLITABU-DT on the problem set given in Archetti et al. [2]. In addition, the authors present 21 new benchmark SDVRP instances with 8 to 288 customers as well as high-quality solutions to these new instances.

2.1.4. Scatter Search. Mota et al. [23] present a new metaheuristic procedure to solve the SDVRP. In particular, they apply scatter search to obtain a low-cost feasible solution that uses the minimum number of routes (i.e., vehicles). This objective function is slightly different from the one minimized in Archetti et al. [2, 3], Boudia et al. [5], and Chen et al. [7].

There may be a number of low-cost solutions using more than the minimum number of vehicles.

A giant tour approach is applied to ensure a set of initial feasible solutions with the minimum number of vehicles. The Clarke and Wright [10] savings algorithm is also used to generate initial feasible solutions. Although there is no guarantee that these solutions minimize the number of vehicles, they often do. These initial feasible solutions are improved using a variety of standard interchange and exchange operations.

A reference set of feasible solutions is then established and rules for combining feasible solutions are specified. If a (new) combined solution is infeasible, it is repaired. If it is feasible, it is improved.

The scatter search results are compared to the results from Archetti et al. [2] over 49 instances ranging in size from 50 to 199 customers. Overall, the scatter search results are not as good as the SPLITABU-DT results. However, because the objective functions are not the same, this comparison is problematic.

2.1.5. Route-Optimization Heuristic Using Mixed-Integer Programming. Archetti et al. [3] model the SDVRP as a route-optimization mixed-integer program. For every feasible route, there is a binary variable in the MIP that determines whether or not a route is part of the optimal solution. It is not computationally tractable to examine all routes to minimize travel cost. The authors develop a technique to identify subsets of routes over which the route-optimization MIP can be solved.

Using the tabu search heuristic given in Archetti et al. [2], edges that appear in a high percentage of solutions are identified, and a set of routes $\bar{R}$ is generated by extending routes through these identified edges. A subset R of $\bar{R}$ is then created according to three criteria: (1) routes of the best-known solution are included, (2) routes with positive value in the solution to the linear programming (LP) relaxation of the route-optimization MIP over $\bar{R}$ are included, and (3) routes with a high desirability are included, where the desirability measures are based on the dual variables of the LP relaxation. A route-optimization phase is then conducted by iteratively generating R and solving the route-optimization MIP.

Archetti et al. [2] track the performance of their route-optimization heuristic and their tabu search heuristic on the data set from Archetti et al. [2]. They also provide the gap between the solution given by the route-optimization heuristic and the LP relaxation solution of the route-optimization MIP over $\bar{R}$ (this is likely to be close to a lower bound for the problem). They find that the average improvement over the tabu search heuristic is approximately 0.5%, and the average gap between the route-optimization solution and the LP relaxation solution is approximately 2.2%.

2.2. Exact Methods and Bound Generating Procedures

2.2.1. Dynamic Programming. Lee et al. [19] examine the multiple vehicle routing problem with split pickups (denoted by mVRPSP). Vehicles with the same capacity are based at a single depot and must pick up items at suppliers and deliver them to the depot. A supplier can be visited by more than one vehicle so that split pickups are permitted.

The authors present two mixed-integer programming formulations of the mVRPSP. They then develop a dynamic programming formulation and show how to solve it using a shortest path approach (SPA).

Lee et al. [19] perform computational tests on a set of 198 small test problems (nine geographic layouts of randomly located suppliers $\times$ 22 supply vectors) where each problem has four, five, or seven suppliers. The SPA solves the problems with four and five suppliers in under a second, whereas the MIP solution times (the MIPs are solved with CPLEX 9.0) range from under a second to more than an hour. The problems with seven suppliers are too large to be solved as MIPs, although SPA solves them all in times that range from under one second to 513 seconds.

2.2.2. Linear Programming with Valid Inequalities. Jin et al. [17] propose a two-stage algorithm to optimally solve the SDVRP. In the first stage, a clustering subproblem is solved in which travel distances are ignored, but all demand is satisfied. At each iteration, this yields a lower bound.

In the second stage, a traveling salesman problem (TSP) is solved for each cluster. Because average customer demand is greater than 10% of vehicle capacity in SDVRPs of interest to the authors, these TSPs are relatively small and easy to solve to optimality. The sum of these TSP lengths over all clusters provides an upper bound on the optimal solution. The authors show how to iterate between the two stages and they develop new valid inequalities for the first stage problem.

Computational experiments demonstrate that on small problems (seven customers), the two-stage algorithm outperforms the dynamic programming approach of Lee et al. [19].

2.2.3. Column Generation. Jin et al. [18] propose a column generation approach to find upper bounds (UB) and lower bounds (LB) for the SDVRP. First, the authors compute the minimum number of vehicles (K) required to satisfy total demand. They allow more than K vehicles in their solution because they seek to minimize the total distance traveled.

Next, they define their master problem and pricing subproblem. The LP relaxation of the master problem is solved using CPLEX 9.0. A limited-search-with-bound algorithm is developed to efficiently solve the pricing subproblem.

The proposed column generation algorithm is tested on 11 problem instances and compared with the cutting plane approach of Belenguer et al. [4]. The column generation algorithm is able to obtain gaps $(UB - LB)/UB$, which are consistently smaller than those generated by Belenguer et al. [4].

2.3. SDVRP Variants

2.3.1. Time Windows. Ho and Haugland [16] study the VRP with time windows and split deliveries (denoted by VRPTWSD). This problem is NP-hard and they show how it can be formulated as a mixed-integer program (they do not try to solve the MIP). The authors develop a three-step heuristic that uses tabu search to solve the VRPTWSD. An initial feasible solution is generated by analyzing travel time and waiting time. This solution is improved by using four different tabu move operators: (1) remove a customer from a route and relocate it to a different route, (2) relocate a customer and split its demand, (3) exchange two customers on different routes, and (4) modified 2-opt exchanges. A postprocessor that is based on unstringing and stringing found in the GENIUS algorithm (Gendreau et al. [13]) is applied to the best solution found during the search process.

Ho and Haugland [16] conduct computational experiments using the six sets of benchmark test problems of Solomon [26]. These problems have 100 customers that are randomly generated, clustered, or semiclustered with Euclidean distances. The demands of the customers are modified by Ho and Haugland [16] so that they could study how the ratio of demand to capacity affects splitting. They solve all problems using their heuristic with splitting and without splitting and report average values for the total distance traveled and the number of vehicles. Ho and Haugland [16] find that, for the most part, splitting deliveries produces better solutions (smaller total distance, fewer vehicles).

2.3.2. Split Deliveries and Pickups (Backhauls). Mitra [21] examines the VRP with split deliveries and pickups (VRPSDP). Deliveries and pickups can occur in any sequence and a customer may be visited more than once by the same vehicle. Mitra [21] first wants to minimize the number of required vehicles and then route the vehicles to minimize the total travel cost. He formulates the VRPSDP as a mixed-integer program and tries to solve problems with 19 customers and one depot to optimality in a reasonable amount of computing time (30 minutes or less). He considers 55 problems with two different sets of edge costs so that there are a total of 110 test problems. Mitra [21] solves 28 problems to optimality and finds upper bounds to the total route cost for the remaining problems.

Mitra [21] develops a heuristic procedure for solving the VRPSDP. The heuristic starts by determining the minimum number of vehicles needed to meet all deliveries and pickups. The routes for the vehicles are then constructed sequentially using cheapest insertion. Mitra [21] reports that the heuristic found 22 of the 28 optimal solutions in about one-fourth of the MIP's computation time, on average.

In a subsequent computational study, Mitra [22] extends his earlier work on the VRPSDP. After a literature review and problem statement, Mitra [22] formulates the VRPSDP as an MIP. Next, he presents a cluster-first, route-second heuristic. The number of clusters is known in advance and is equal to the minimum number of vehicles required. The expression for this number is given in Mitra [22].

Once clusters are formed, a route construction procedure is applied. Next, the proposed heuristic is compared with the author's earlier heuristic (see Mitra [21]) over a problem set of 110 instances. The new heuristic is found to perform statistically better than the earlier heuristic. Finally, Mitra [22] applies his cluster-first, route-second heuristic to the VRP with simultaneous deliveries and pickups, runs some preliminary computational experiments, and compares his results to those found in Chen and Wu [6].

2.3.3. Heterogeneous Fleet. Tavakkoli-Moghaddam et al. [28] consider the capacitated vehicle routing problem with split services and a heterogenous fleet, denoted by CHVRPSS. In the CHVRPSS, there are Q vehicle classes each with a different capacity. Each class q contains v_q vehicles, $q = 1, \ldots, Q$, and the total number of available vehicles is $V = \sum_{q=1}^{Q} v_q$. The authors formulate the CHVRPSS as a mixed-integer program. They use an objective function that contains travel cost, a cost per vehicle, and a penalty term for unused capacity. The MIP is solved to optimality using Lingo 8.0 on five instances with six nodes.

Tavakkoli-Moghaddam et al. [28] propose a simulate annealing (SA) heuristic for solving the CHVRPSS. An initial solution is generated by considering vehicles ordered by capacity (largest to smallest) and adding random customers to a current route until capacity is reached. Next, the simulated annealing heuristic is applied to the problem. In the SA heuristic, a neighbor of a current solution is explored through either a one-node move or a two-node move (the move selection is random). The neighbor replaces the current solution with a probability that depends on the cost difference between the neighbor and the current solution, and the temperature of the algorithm. After a fixed number of iterations at different temperatures, the heuristic stops and returns the best solution.

Tavakkoli-Moghaddam et al. [28] test the SA heuristic on the five instances solved to optimality. They find an average gap of 1.36%. On 19 larger problems (10 to 100 nodes), they compare the results of the SA heuristic to a lower bound obtained by solving a traveling salesman problem on all nodes. On average, the SA heuristic is approximately 26% above the lower bound.

2.3.4. Real-Time Events. Thangiah et al. [29] study the split-delivery pickup and delivery time window problem with transfers (SDPDTWP) over a real-time horizon. In the SDPDTWP, a fleet of uncapacitated vehicles must deliver a set of shipments. Each shipment has a time window $[a, b]$, where a is the earliest time a shipment can be picked up from its origin, and b is the latest time a shipment can be dropped off at its destination. A shipment can be split or transferred to reduce travel time. A split occurs when shipments from the same origin are serviced by different vehicles. A transfer occurs when a vehicle leaves a shipment at an intermediate stop to be picked up and delivered to its destination by a different vehicle. There is no central depot in the SDPDTWP. A vehicle begins its route at one of the origin nodes and ends its route at the last destination node on its route. The SDPDTWP is set in real-time. Events including deletion, insertion, and modification of a shipment, and deletion (breakdown) and insertion of a vehicle can occur throughout the time horizon of an SDPDTWP instance.

There are three objectives in the SDPDTWP. First, minimize the number of vehicles that is needed to make all pickups and deliveries within the specified time windows. The number of available vehicles must be determined a priori (before any routing is done), so it might not be possible to make all pickups and deliveries using the available fleet. Second, minimize the number of shipments that need to be rescheduled (these are the shipments that cannot be serviced during their time windows). Third, minimize the total travel time of the fleet.

Thangiah et al. [29] develop a heuristic for solving the SDPDTWP that is based on the work of Shang and Cuff [25]. First, the number of vehicles is determined and shipments are inserted into routes. Second, a local search is conducted. When real-time events occur throughout the time horizon, the heuristic will respond. For example, when a new shipment is introduced, the heuristic attempts to reroute the vehicles in a way that includes the new shipment while minimizing the effects on travel time and other shipments.

Thangiah et al. [29] test their heuristic on a static instance (no real-time events) of 159 shipments given by Shang and Cuff [25]. For this instance, the authors' heuristic produces a solution that services all shipments while using fewer vehicles and reducing average travel time by over 75% when compared with the results of Shang and Cuff [25]. They also test their heuristic on new instances that incorporate real-time events into the first instance. The authors examine how different real-time events affect the routes in terms of the number of unserviced customers and travel times.

2.3.5. Delivering Multiple Products on a Fixed Route. Liu et al. [20] examine a variant of the SDVRP in which multiple products are delivered to a set of customers on a fixed route. They call this problem the multiproduct packing-delivery problem with a fixed route, and denote the problem by P. In P, n customers have a fixed order $1,\dots,n$ in which they must be visited. There are K products of varying size per unit demand, and each customer has a demand for each product. The objective is to partition the customers along the fixed sequence into feasible trips (i.e., trips that do not violate vehicle capacity) in a way that minimizes total travel cost. Service at a customer can be split between the last stop on a trip (the stop immediately preceding the return to the depot) and the first stop on the following trip.

Liu et al. [20] develop a heuristic for solving P. First, the optimal solution is found for the nonsplit case p_0 over a sequence of customers S (initially S is the entire sequence of customers). The authors show that, by converting the nonsplit problem to a shortest path problem, p_0 can be found in $O(n^2\log(n))$ time. Next, the first trip in p_0 with excess capacity is extended to use all of its leftover capacity, and this leads to a split delivery for a customer j. Let j^- represent customer j on the current route and j^+ represent customer j on the next route. A bin-packing routine determines the products to be delivered to node j^-. Optimal nonsplit solutions are determined for the segments $S_1=\{1,2,\dots,j^-\}$ and $S_2=\{j^+,\dots,n\}$. If the sum of the costs of these two solutions is less than the cost of p_0, then the split is made. The solution from segment S_1 is added to the end of the current solution p and the process repeats with S_2 replacing S. If the sum of the costs is greater than the cost of p_0, then the split is not made and the next candidate split is considered. If there are no candidate splits to consider, then the algorithm adds p_0 to the end of p, stops, and returns p.

Liu et al. [20] test their heuristics on instances with one, two, or three products of varying size. The instances have 50, 100, 200, 300, or 400 customers randomly generated in a 100 by 100 square in the plane. Demands all fall into a range of [5%–15%], [10%–40%], [0%–100%], or [25%–75%] of vehicle capacity, and are a random mix of the K products. A total of 14,000 different instances were used, and the authors provide graphs illustrating that improvement in solution quality, usually between 8% and 12%, can be achieved by splitting deliveries.

2.3.6. Pickup and Delivery with Split Loads. Nowak et al. [24] introduce the pickup and delivery problem with split loads (PDPSL). The PDPSL is modeled on a network of load origin nodes, load destination nodes (a node may serve as both an origin and a destination node), edges with travel costs between the nodes, and transportation requests. A transportation request is a load of goods that must be delivered from a specific origin to a specific destination. When a vehicle arrives at an origin, it can pick up any amount of any load to be delivered, up to its capacity. When a vehicle arrives at a destination, it delivers the destination's entire load. The objective of the PDPSL is to find a route for a single vehicle that meets all transportation requests while minimizing travel costs. There is no depot in the PDPSL. A vehicle begins its route at a prespecified origin node and can end at any of the destination nodes.

Nowak et al. [24] develop a heuristic for solving the PDPSL with three basic steps. First, they generate an initial solution that has dedicated trips directly from the origin to the destination for each transportation request. Second, feasible splits are identified by comparing load sizes to occurrences of excess capacity along a vehicle's route. Then a split is made with a probability determined by the profitability of the split (e.g., a very profitable split has a high probability of being accepted). After a split is made, it is added to a tabu list to ensure that it is not subsequently undone, or selected again. Third, improvement procedures including a route combination routine similar to the Clarke and Wright [10] algorithm for the VRP, a load swap routine, and a load insertion routine are applied, and the best solution is saved.

The authors test their heuristic on two problem sets, each with 120 instances of three sizes: small (5 origins, 15 destinations, and 75 requests), medium (5 origins, 20 destinations, and 100 requests), and large (5 origins, 25 destinations, and 125 requests). Origin and destination locations and load sizes are randomly generated within specified ranges. The authors also test their heuristic with the splitting step omitted on the same instances. They observe that in most instances significant savings are achieved by allowing split deliveries.

Nowak et al. [24] provide the results of a computational experiment performed for an anonymous third-party logistic provider (3PL). To meet the real-world requirements of the 3PL, the PDPSL is modified in several ways: using a multivehicle fleet, penalizing one-way trips, enforcing minimum and maximum tour lengths, and imposing a financial and time-associated cost for each stop. Their heuristic for the PDPSL was modified and run with and without splits on the 3PL data. A very modest savings of approximately 1% on average was achieved by splitting. The authors attribute the low level of savings to the complexity added by the real-world constraints.

3. Computational Issues for the SDVRP

In this section, we discuss computational issues for the standard SDVRP.

3.1. Problem Sets

There are three sets of benchmark problems for the SDVRP. Several papers focus on the six problems $(1, 2, 4, 5, 11, 12)$ from Christofides and Eilon [8] and Christofides et al. [9] with 50, 75, 100, 120, 150, and 199 customers. For each problem, a customer's demand is generated according to the six scenarios ([0.01–0.1], [0.1–0.3], [0.1–0.5], [0.1–0.9], [0.3–0.7], [0.7–0.9]) given by Dror and Trudeau [11]. The demand for customer i in scenario $[\alpha - \beta]$ with a vehicle capacity of k units is randomly selected from a uniform distribution on the interval $[\alpha k, \beta k]$ (we denote this problem set by CEMT).

Belenguer et al. [4] develop a set of 14 random problems with 50, 75, and 100 customers where the vehicle capacity is 160 and the six scenarios of Dror and Trudeau [11] are used to randomly generate a customer's demand (we denote this problem set by BMM). All problems are available online at http://www.uv.es/belengue/sdvrp.html.

Recently, Chen et al. [7] developed 21 test problems that range in size from eight to 288 customers (we denote this problem set by CGW). Vehicle capacity is 100 units and customer demand is either 60 or 90 units. The problems are generated along the lines of Scenario 6 with very large customer demand ([0.7–0.9]) from Dror and Trudeau [11]. Each problem has a geometric symmetry (star shape) with customers located in concentric circles around the depot that allows the authors to visually estimate a near-optimal solution. The problem generator and specifications are given in Chen et al. [7]. These problems are available online at http://www.rhsmith.umd.edu/faculty/bgolden/index.html.

3.2. Reporting Computational Results

It is not a straightforward task to compare results across different papers. For example, consider the results reported by Archetti et al. [2] and Chen et al. [7].

Archetti et al. [2] randomly generate problems from CEMT. They use six demand scenarios and six problem sizes (50 to 199 customers) for a total of $6 \times 6 = 36$ problems. They run SPLITABU-DT five times on one instance of each scenario and provide the average percent improvement over Dror and Trudeau's [11] results. The authors use a 2.4 GHz Pentium 4 processor with 256 MB of RAM.

Chen et al. [7] randomly generate 36 problems from CEMT with the same demand scenarios and problem sizes used by Archetti et al. [2]. For each problem, they solve 30 instances on a 1.7 GHz Pentium 4 processor and 512 MB of RAM. How do you compare the results reported in both papers?

The first key issue is that the 36 problems are randomly generated, making them different in both papers. Any direct comparison is flawed from the outset. You might be thinking: Why didn't Chen et al. [7] use the 36 actual problem instances solved by Archetti et al. [2] in their 2006 paper? Well, they were simply not available.

Let's take the reported results for a 50-customer problem with the [0.01–0.1] scenario. Archetti et al. [2] solve one instance of this problem five times and report an average improvement of 5.12% over Dror and Trudeau's [11] algorithm. Chen et al. [7] solved 30 different instances of this problem and have 30 solution values. How do you compare the 5.12% to the 30 solution values?

Archetti et al. [2] were kind enough to provide Chen et al. [7] with the actual solutions produced by SPLITABU-DT for each problem size and scenario. These solutions are shown in Table 2.

These are more detailed results, but it is still not easy to make a direct comparison of five solution values to 30 solution values. The second key issue is: How do you make reasonable comparisons when the data are different?

Chen et al. [7] propose a simple statistical test. If EMIP + VRTR and SPLITABU-DT are equally good with respect to solution quality, then SPLITABU-DT would beat the

Table 2. Median values produced by EMIP + VRTR and actual values produced by SPLITABU-DT for a problem from CEMT.

50 customers with vehicle capacity 160		SPLITABU-DT				
Scenario	EMIP + VRTR*	1	2	3	4	5
[0.01–0.1]	457.21	464.64	464.64	466.19	460.79	462.54
[0.1–0.3]	723.57	751.60	767.46	752.84	760.57	774.56
[0.1–0.5]	943.86	1,013.00	1,015.15	997.22	1,007.13	1,010.86
[0.1–0.9]	1,408.34	1,461.01	1,473.29	1,470.11	1,443.84	1,501.39
[0.3–0.7]	1,408.68	1,507.60	1,491.92	1,490.73	1,487.02	1,507.25
[0.7–0.9]	2,056.01	2,166.34	2,174.81	2,166.11	2,170.43	2,148.38

*Median solution value over 30 instances given in Chen et al. [7].

median EMIP + VRTR result about half the time. Using a binomial distribution with $n = 36$ (this corresponds to one run over 36 cases) and $p = 1/2$, they test the null hypothesis that the results of the two methods are equally good (H_0: $p = 0.50$) against the alternative hypothesis that SPLITABU-DT performs worse than the median value of EMIP + VRTR (H_a: $p < 0.50$). Using a significance level of 0.01, the null hypothesis would be rejected when $(\hat{p} - 0.5)/\sqrt{(0.5)(0.5)/36} \leq -2.33$ or $\hat{p} \leq 0.3058$. If SPLITABU-DT performs better than the median value of EMIP + VRTR in $(0.3058)(36) = 11$ instances or less for a single run over 36 cases, then the null hypothesis would be rejected. The median values for EMIP + VRTR are given in Table 2.

For each of the five runs of SPLITABU-DT over the 36 cases, Chen et al. [7] count the number of times the SPLITABU-DT solution is better than the median solution of EMIP + VRTR. For each run, the count for SPLITABU-DT is much less than 11, and therefore, they reject the null hypothesis and conclude that SPLITABU-DT performs worse than EMIP + VRTR.

3.3. Summary of Computational Issues

The availability of only randomly generated problems coupled with different computing platforms makes the comparison of published computational results difficult. Most algorithms have not been run on exactly the same set of SDVRP test problems. Sometimes researchers devised clever tests or used comparison contortions to make their points. Larger problems with visually estimated solutions are now available for researchers to test their algorithms.

4. Conclusions and Future Directions

In recent years, the SDVRP has drawn a significant amount of attention in the operations research literature. Powerful solution methods including tabu search, simulated annealing, record-to-record travel algorithm, genetic algorithms, dynamic programming, and mixed-integer programming have been applied to the SDVRP and have produced high-quality results to benchmark problems. Researchers have begun to consider interesting variants of the SDVRP that account for time windows, pickups, and backhauls.

In the future, we expect that, with rapidly rising fuel prices and increasing vehicle purchase and maintenance costs, companies with significant routing components will try to reduce costs by considering split deliveries. Operations research practitioners and researchers will play an important role in developing the algorithms for implementation in software and systems for solving practical applications of the SDVRP.

References

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

[2] C. Archetti, M. Speranza, and A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science* 40:64–73, 2006.

[3] C. Archetti, M. Speranza, and M. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science* 42:22–31, 2008.

[4] J. Belenguer, M. Martinez, and E. Mota. A lower bound for the split delivery vehicle routing problem. *Operations Research* 48:801–810, 2000.

[5] M. Boudia, C. Prins, and M. Reghioui. An effective memetic algorithm with population management for the split delivery vehicle routing problem. T. Bartz-Beielstein, M. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, eds. *Proceedings of the Fourth Annual International Workshop on Hybrid Metaheuristics HM (2007), Lecture Notes in Computer Science*, Vol. 4771. Springer, Berlin, 16–30, 2007.

[6] J. Chen and T. Wu. Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society* 57:579–587, 2006.

[7] S. Chen, B. Golden, and E. Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks* 49:318–329, 2007.

[8] N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly* 20:309–318, 1969.

[9] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, eds. *Combinatorial Optimization*. John Wiley & Sons, Chichester, UK, 315–338, 1979.

[10] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12:568–581, 1964.

[11] M. Dror and P. Trudeau. Savings by split delivery routing. *Transportation Science* 23:141–145, 1989.

[12] M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research* 40:1086–1094, 1992.

[13] M. Gendreau, A. Hertz, G. Laporte, and M. Stan. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research* 46:330–335, 1998.

[14] B. Gillett and L. Miller. A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 22:340–349, 1974.

[15] B. Golden, S. Raghavan, and E. Wasil, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, 2008.

[16] S. Ho and D. Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers and Operations Research* 31:1947–1964, 2004.

[17] M. Jin, K. Liu, and R. Bowden. A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem. *International Journal of Production Economics* 105:228–242, 2007.

[18] M. Jin, K. Liu, and B. Eksioglu. A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters* 36:265–270, 2008.

[19] C. Lee, M. Epelman, C. White, III, and Y. Bozer. A shortest path approach to the multiple-vehicle routing problem with split pick-ups. *Transportation Research Part B*: *Methodological* 40:265–284, 2006.

[20] S. Liu, L. Lei, and S. Park. On the multi-product packing-delivery problem with a fixed route. *Transportation Research Part E*: *Logistics and Transportation Review* 44:350–360, 2008.

[21] S. Mitra. An algorithm for the generalized vehicle routing problem with backhauling. *Asia-Pacific Journal of Operational Research* 22:153–169, 2005.

[22] S. Mitra. A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of Operational Research Society* (ePub ahead of print 19 September 2007) 1–15.

[23] E. Mota, V. Campos, and Á. Corberán. A new metaheuristic for the vehicle routing problem with split deliveries. C. Cotta and J. van Hemert, eds. *Lecture Notes in Computer Science 4446: Evolutionary Computation in Combinatorial Optimization*. Springer, Berlin, 121–129, 2007.

[24] M. Nowak, Ö. Ergun, and C. White, III. Pickup and delivery with split loads. *Transportation Science* 42:32–43, 2008.

[25] J. Shang and C. Cuff. Multicriteria pickup and delivery problem with transfer opportunity. *Journal Computers and Industrial Engineering* 30:631–645, 1996.

[26] M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research* 35:254–265, 1987.

[27] K. Sörensen and M. Sevaux. MA|PM: Memetic algorithms with population management. *Computers and Operations Research* 33:1214–1225, 2006.

[28] R. Tavakkoli-Moghaddam, N. Safaei, M. Kah, and M. Rabbani. A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute* 344:406–425, 2007.

[29] S. Thangiah, A. Fergany, and S. Awan. Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operational Research* 15:329–349, 2007.