Computers & Industrial Engineering 61 (2011) 794–804

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

The multi-depot split delivery vehicle routing problem: An integer programming-based heuristic, new test problems, and computational results

Damon Gulczynski [a], Bruce Golden [b], Edward Wasil [c,*]

[a] Department of Mathematics, University of Maryland, College Park, MD 20742, United States
[b] Robert H. Smith School of Business, University of Maryland, College Park, MD 20742, United States
[c] Kogod School of Business, American University, Washington, DC 20016, United States

ARTICLE INFO

Article history:
Received 22 August 2010
Received in revised form 1 May 2011
Accepted 23 May 2011
Available online 30 May 2011

Keywords:
Multi-depot
Split delivery
Vehicle routing problem
Integer program

ABSTRACT

The multi-depot split delivery vehicle routing problem combines the split delivery vehicle routing problem and the multiple depot vehicle routing problem. We define this new problem and develop an integer programming-based heuristic for it. We apply our heuristic to 30 instances to determine the reduction in distance traveled that can be achieved by allowing split deliveries among vehicles based at the same depot and vehicles based at different depots. We generate new test instances with high-quality, visually estimated solutions and report results on these instances.

1. Introduction

In the traditional vehicle routing problem (VRP), vehicles are dispatched from a single depot to service the demands of customers. In the multi-depot VRP (MDVRP), vehicles are dispatched from several depots, while in the split delivery VRP (SDVRP) a customer's demand can be split among vehicles on different routes. There is a large literature devoted to modeling these two variants of the VRP, and we review relevant papers in Section 2.

Recently, we have become aware of problems arising in real-world settings such as waste collection and inventory distribution that can combine the MDVRP and the SDVRP. In major cities in Taiwan (Chao, 2008), residents bring their waste to one of several neighborhood collection sites. Vehicles transport the waste from the collection sites to permanent landfills. A collection site can be visited by vehicles two or more times in a day. By treating the collection sites as customers with demands equal to their expected daily amounts of waste, a collection site (customer) does not need to have all of its demand satisfied by one vehicle on a single visit (a visit to a site can be split among several vehicles) and there are multiple landfills (depots) for the vehicles to dump their waste.

In inventory distribution, Li, Chu, and Chen (2011) consider a problem where the inventory of a retailer can be replenished by splitting the delivery among several vehicles. Although the authors model a one-warehouse distribution system, multiple warehouses (depots) are commonly used in practice to service customers (Wang, Fung, & Chai, 2004).

The combination of multiple depots and split deliveries can also arise in disaster relief logistics. Consider a disaster similar in scope to Hurricane Katrina, which devastated the coasts of Louisiana and Mississippi in 2005. Suppose another such hurricane hits the same southern coastline of the US. There will be an immediate demand for emergency supplies such as meals ready to eat (MREs), blankets, cots, flashlights, tents, sleeping bags, water, and medical supplies. These supplies will be sent (most likely) from two of the eight FEMA logistics centers in the continental US. These are in Atlanta, GA and Ft. Worth, TX. Multiple (split) deliveries will be made to each demand location. See Afshar (2011) for details.

We propose a new variant – the multi-depot split delivery vehicle routing problem (MDSDVRP) – that models routing considerations encountered in practice. The objective in the MDSDVRP is to minimize the total distance traveled by the fleet across all depots, while allowing more than one vehicle to satisfy a customer's demand.

Let $V=\{v_1,\ldots,v_N\}$ be the set of customers and let $W=\{w_1,\ldots,w_M\}$ be the set of depots. Let D_i be the demand of customer i, and let Q be the capacity of a vehicle. Let the distance between a pair of nodes $e=(i,j)$ be denoted by c_e (or c_{ij}). Given route r, let $V(r)$ be the set of nodes and $E(r)$ the set of travel edges on r. Let d_{ir} be the amount delivered to customer i on r. We want to find a set of routes R such that

* Corresponding author.
E-mail addresses: damon@math.umd.edu (D. Gulczynski), bgolden@rhsmith.umd.edu (B. Golden), ewasil@american.edu (E. Wasil).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.05.012

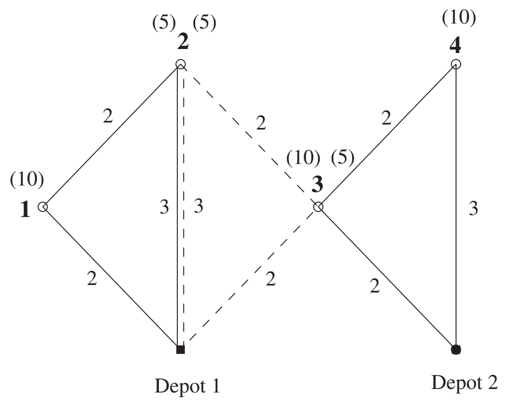


Fig. 1. In this MDSDVRP, there are four customers and two depots. Node labels in parentheses are delivery amounts and edge labels are distances. The vehicle capacity is 15. In this solution, the total distance traveled is 21 units.

- route r begins and ends at w_k, for some $k \in \{1, \ldots, M\}$, for all $r \in R$ (a route starts and ends at the same depot),
- $\sum_{i \in V(r)} d_{ir} \leqslant Q$, for all $r \in R$ (vehicle capacity restriction),
- $\sum_{r \in R} d_{ir} = D_i$, for all $i \in V$ (demand must be satisfied for each customer),
- minimize $\sum_{r \in R} \sum_{e \in E(r)} c_e$ (total distance is minimized).

In Fig. 1, we show an MDSDVRP. We have four customers (nodes 1–4) and two depots. Edge labels are distances and node labels in parentheses are delivery amounts. Customers 1, 2, and 4 have a demand of 10 units, and customer 3 has a demand of 15 units. The vehicle capacity is 15 units. There are three routes: two starting and ending at depot 1 (indicated by solid and dashed lines) and one starting and ending at depot 2. The delivery at customer 2 is split between routes using depot 1. The delivery at customer 3 is split between a route using depot 1 and a route using depot 2. The total distance traveled is 21 units.

In this paper, we develop an integer-programming based heuristic for the MDSDVRP. Typically, heuristics in the vehicle routing literature use some form of local search as part of a metaheuristic (such as tabu search or a genetic algorithm). In these heuristics, many small solution neighborhoods are explored. Recently, De Franceschi, Fischetti, and Toth (2006), Chen, Golden, and Wasil (2007) and Gulczynski, Golden, and Wasil (2010) have taken a different approach. They developed integer programming-based heuristics that explore relatively few large solution neighborhoods. The results generated by their heuristics demonstrate that integer programming can be an effective tool in generating high-quality solutions to vehicle routing problems. We continue along these lines by developing an integer programming-based heuristic for a new problem in the vehicle routing literature.

The remainder of this paper is organized as follows. In Section 2, we review the literature on the MDVRP and the SDVRP. In Section 3, we develop an integer programming-based heuristic for the MDSDVRP. In Section 4, we present computational results. In Section 5, we give our conclusions.

2. Literature review of the MDVRP and the SDVRP

The literature for the MDVRP dates back over 35 years, and the literature for the SDVRP dates back over 20 years. To our knowledge, we are the first researchers to consider the MDSDVRP.

In the 1970s and 1980s, heuristics for the MDVRP were developed by Tillman and Cain (1972), Wren and Holliday (1972), Gillett and Johnson (1976), Golden, Magnanti, and Nguyen (1977) and Raft (1982). In the 1990s, Chao, Golden, and Wasil (1993) developed a record-to-record travel algorithm for the MDVRP. Renaud, Laporte, and Boctor (1996) and Cordeau, Gendreau, and Laporte (1997) used tabu search. In 2001, Thangiah and Salhi (2001) developed a genetic clustering heuristic. In 2007, Pisinger and Røpke (2007) applied an adaptive large neighborhood search algorithm to several vehicle routing problems, including the MDVRP.

The first heuristics for the SDVRP were developed by Dror and Trudeau (1989) and Dror and Trudeau (1990) in the late 1980s. They used a two-stage algorithm that incorporated k-split interchanges and route additions. Recently, Archetti, Speranza, and Hertz (2006) developed a tabu search algorithm. Chen et al. (2007) combined an endpoint mixed integer program and a variable length record-to-record travel algorithm. Mota, Campos, and Corberán (2007) used scatter-search. Gulczynski et al. (2010) extended the work of Chen et al. (2007) by developing a heuristic for the SDVRP with minimum delivery amounts. Derigs, Li, and Vogel (2009) applied a local search-based metaheuristic to the SDVRP. The papers by Archetti and Speranza (2008) and Gulczynski, Golden, and Wasil (2008) are good sources for recent developments in modeling and solving the SDVRP.

Researchers have considered several variants of the MDVRP and the SDVRP that involve time windows (Cordeau, Laporte, & Mercier, 2004; Giosa, Tansini, & Viera, 2002; Ho & Haugland, 2004; Polacek, Hartl, Doerner, & Reimann, 2004) and pickups (backhauls) (Mitra, 2005; Salhi & Nagy, 1999). Two practical applications of the MDVRP and the SDVRP are delivering groceries (Crevier, Cordeau, & Laporte, 2007; Golden & Wasil, 1987) and collecting waste (Angelelli & Speranza, 2002; Archetti & Speranza, 2004).

3. An integer programming-based heuristic for the MDSDVRP

3.1. Assigning customers to depots

We describe a heuristic for the MDSDVRP. First, we assign customers to depots using a procedure developed by Golden et al. (1977). For each customer i, we let λ_i be the distance between i and the closest depot to i and λ'_i be the distance between i and the second closest depot to i. If $\frac{\lambda_i}{\lambda'_i}$ is less than a tolerance ϵ, then customer i is immediately assigned to its closest depot. If $\frac{\lambda_i}{\lambda'_i} \geqslant \epsilon$, then i is left unassigned temporarily. In this way, a customer that is much closer to one depot than other depots will be immediately assigned to its closest depot. A customer that is nearly equidistant from multiple depots will be assigned using cheapest insertion.

After the initial assignment phase, unassigned customers are assigned to depots based on a cheapest insertion criterion. For each unassigned customer i and each depot w, we calculate the cost of inserting i between each pair of customers already assigned to w (we consider w as a customer assigned to itself). Then we assign i to the same depot as the pair giving the cheapest insertion. That is, we assign customer i to the same depot as customers j and k where $c_{ij} + c_{ik} - c_{jk}$ is the smallest value over all pairs of customers already assigned to a depot.

3.2. Solving the SDVRP on each depot separately

After all customers have been assigned to depots, we solve the SDVRP on each depot and its assigned customers separately using the EMIP-MDA + ERTR heuristic developed by Gulczynski et al. (2010).

EMIP-MDA + ERTR is a two-stage heuristic that improves an initial solution to the VRP with no splits. The initial solution is generated using a modified Clarke–Wright (CW) algorithm (Yellow, 1970). In the first stage, an endpoint mixed integer program with minimum delivery amounts (EMIP-MDA) is formulated and solved (this formulation is based on the EMIP model developed by Chen et al. (2007)). The EMIP-MDA maximizes the savings from splitting deliveries at certain customers and reallocating some (or all) of their demands to new routes. A time limit T is set for the EMIP-MDA. If a solution is not returned after T seconds, EMIP-MDA terminates, and the best solution found up to that point is returned.

In the second stage, the enhanced record-to-record travel algorithm (ERTR) is applied to reduce the total distance traveled by the fleet. ERTR is a heuristic for the VRP that does not produce any new split deliveries. It is a modified version of the variable length record-to-record travel algorithm (VRTR) developed by Li, Golden, and Wasil (2005). VRTR improves a VRP solution by performing one-point and two-point node exchanges, as well as two-opt edge exchanges. ERTR was developed by Groër, Golden, and Wasil (2010) and uses additional moves such as three-point node exchanges and Or-opt edge exchanges. Uphill moves are allowed when a solution is within a preset tolerance of the record solution. The details of ERTR are given in Table 1. In Table 2, we give an outline of EMIP-MDA + ERTR. A complete description can be found in Gulczynski et al. (2010).

3.3. Formulating the MDSDVRP as a mixed integer program

By applying EMIP-MDA + ERTR to each depot and the customers assigned to it, we generate an initial solution to the MDSDVRP. We denote the initial solution by S. In S, there are no deliveries split among vehicles from different depots. We now describe a mixed integer program (MIP) that attempts to improve S by considering additional split deliveries, including inter-depot split deliveries.

For each customer i on each route r, we calculate the cost of inserting i immediately prior to customer j on route q, where q does not begin and end at the same depot as r. Let φ denote the minimum insertion cost of i across all j and q. For each route, we add the two customers with the smallest values of φ to the inter-depot candidate set denoted by ID. If a route has only one customer, it is added to ID. The customers in ID are called id-nodes.

Table 1
Enhanced record-to-record travel algorithm (ERTR) for the VRP.

S_d = VRP solution with depot d, *deviation* = 1%, *count* = 0, L = 10, K = 70
Initialize the record, $S_d^* = S_d$
Uphill:
For (k = 1 to K)
  Apply one-point moves, apply two-point moves, apply three-point moves
  Apply two-opt edge exchanges, apply Or-opt edge exchanges
  Update S_d if result is within *deviation* of cost of S_d^*
  Update S_d^* if necessary
End-For
Downhill:
Apply Or-opt edge exchanges, apply two-opt edge exchanges
Apply one-point moves, apply two-point moves, apply three-point moves
If (cost decreases), update S_d and go to Downhill
Else Update S_d^* if necessary, otherwise *count* = *count* + 1
If (*count* < L), go to Uphill
Perturb solutions once and go to Uphill
Return S_d^*

Table 2
EMIP-MDA + ERTR algorithm for the SDVRP.

Generate initial VRP solution S (no split deliveries) using modified Clarke–Wright algorithm
Apply EMIP-MDA to S to generate improved solution S_1 (make distance-reducing split deliveries)
Apply ERTR to S_1 to generate improved solution S^* (sequence customers efficiently on routes)
Return S^*

Let $NE(i)$ be the neighborhood of customer $i \in ID$. This neighborhood is the set of id-nodes j for which c_{ij} is the smallest. Each neighborhood contains L customers, where L is a preset parameter. We consider moving some (or all) of the demands serviced at id-nodes to new locations. For each $i \in ID$ and j in $NE(i)$, there are three possible moves: (1) move all of the demand of i immediately prior to j, (2) move some of the demand of i immediately prior to j (split i's delivery), and (3) move none of the demand of i immediately prior to j. Let $p(v)$ and $s(v)$ be the predecessor and successor of customer v, respectively. The savings associated with the three possible moves are: (1) $-c_{p(j)i} - c_{ij} - c_{p(i)s(i)} + c_{p(j)j} + c_{p(i)i} + c_{is(i)}$, (2) $-c_{p(j)i} - c_{ij} + c_{p(j)j}$, and (3) zero. In Fig. 2, we show the possible moves.

In order to optimally reallocate demands across all id-nodes, we formulate an inter-depot mixed integer program (IDMIP) that is based on the EMIP-MDA. In the IDMIP, let R be the set of routes of solution S. Let Q_r be the residual capacity of route $r \in R$ (that is, the vehicle capacity minus the total amount delivered on r). Let D_i be the demand of id-node i.

The decision variables are defined as follows. Let b_i equal 1 if all of id-node i's demand is moved (that is, i is removed from its current route), and 0 otherwise; m_{ij} equals 1 if id-node i is inserted before $j \in NE(i)$, and 0 otherwise; and d_{ij} is the amount of id-node i's demand that is moved before $j \in NE(i)$.

Our formulation of the IDMIP is given by the following objective function and constraints.

$$\text{maximize} \quad \sum_{i \in ID} b_i(c_{p(i)i} + c_{is(i)} - c_{p(i)s(i)}) - \sum_{i \in ID} \sum_{j \in NE(i)} m_{ij}(c_{p(j)i} + c_{ij} - c_{p(j)j}) \tag{1}$$

$$\text{subject to} \quad \sum_{i:j \in NE(i)} d_{ij} + \sum_{q:k \in NE(q)} d_{qk} - \sum_{l \in NE(k)} d_{kl} - \sum_{t \in NE(j)} d_{jt} \leqslant Q_r \quad \forall r \in R; k,j \text{ id-nodes of route } r \tag{2}$$

$$\sum_{j \in NE(i)} d_{ij} \leqslant D_i \qquad \forall i \in ID \tag{3}$$

$$\sum_{j \in NE(i)} d_{ij} \geqslant D_i b_i \qquad \forall i \in ID \tag{4}$$

$$D_i m_{ij} \geqslant d_{ij} \qquad \forall i \in ID, \quad \forall j \in NE(i) \tag{5}$$

$$1 - b_i \geqslant \sum_{j:i \in NE(j)} m_{ji} \qquad \forall i \in ID \tag{6}$$

$$1 - b_{p(i)} \geqslant \sum_{j:i \in NE(j)} m_{ji} \qquad \forall i \in ID \tag{7}$$

$$b_k + b_{p(k)} \leqslant 1 \qquad \forall r \in R; \; k, p(k) \text{ id-nodes of route } r \tag{8}$$

$$d_{ij} \geqslant 0 \qquad \forall i \in ID, \quad \forall j \in NE(i) \tag{9}$$

$$m_{ij}, b_i \in \{0, 1\} \qquad \forall i \in ID, \quad \forall j \in NE(i) \tag{10}$$

In the objective function (1), we maximize the total savings across all possible id-node moves. Constraints (2) ensure that feasibility is maintained with respect to vehicle capacity, that is, the total amount of demand moved to a route minus the total amount moved from a route is less than the residual capacity. In constraints (3), we cannot move more than the actual demand at a customer. In constraints (4), we ensure that all demand is moved from customer i if $b_i = 1$, while in constraints (5), we ensure that $m_{ij} = 1$ if any demand is moved from customer i prior to customer j. Constraints (6) and (7) prevent any demand from being moved prior to customer i if i or $p(i)$ has been removed from its route. In constraints (8), we eliminate the possibility of removing both customers k and $p(k)$. Constraints (6)–(8) ensure the objective value reflects the savings accurately, as the coefficients in the objective function

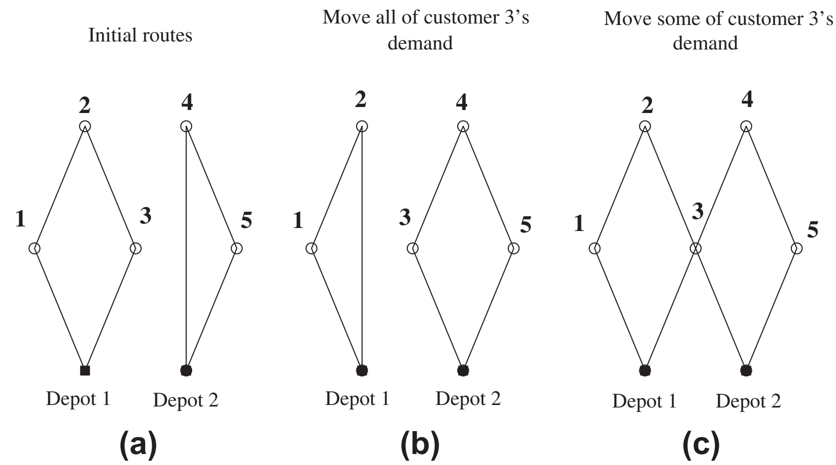


Fig. 2. Three possible customer moves.

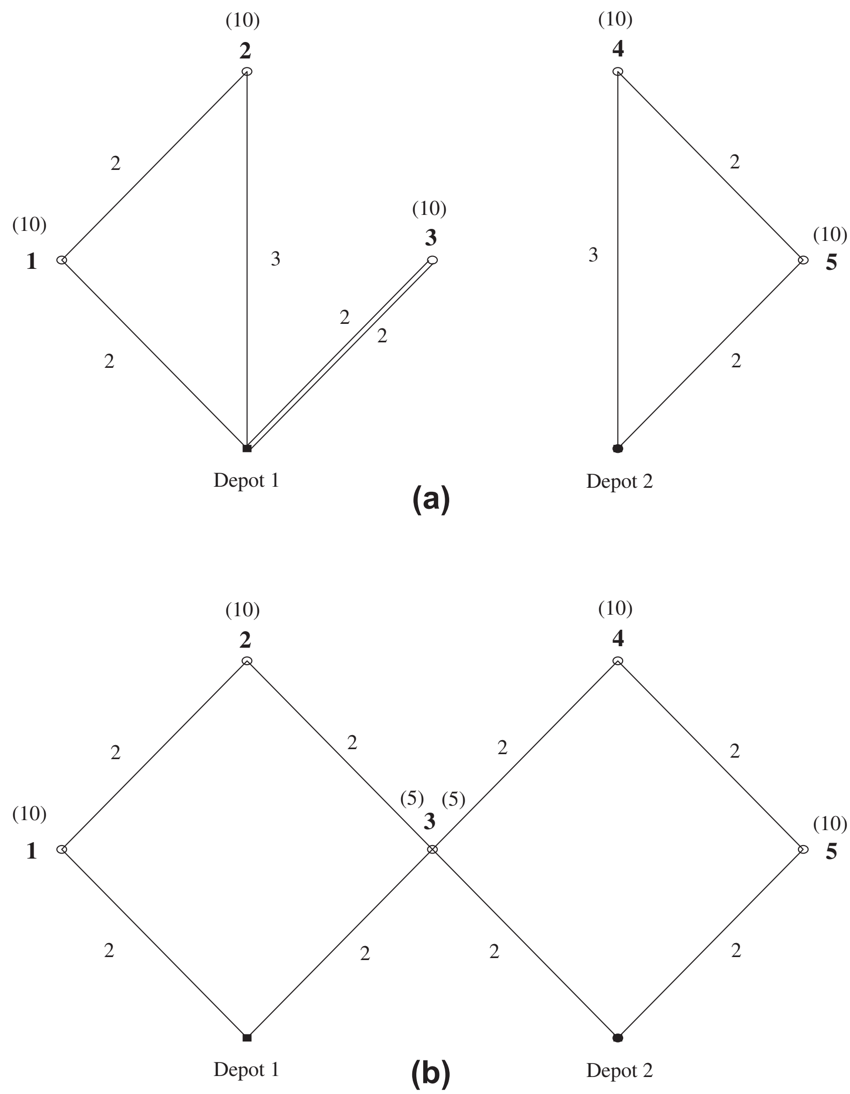


Fig. 3. In the solution shown in (a), there are five customers and two depots. Node labels in parentheses are delivery amounts and edge labels are distances. The vehicle capacity is 25 and the total distance traveled is 18 units. In the improved solution produced by IDMIP shown in (b), the total distance traveled is 16 units.

depend on a customer and its predecessor. Constraints (9) and (10) ensure nonnegativity and 0–1 solutions.

To illustrate, we apply IDMIP to the solution given in Fig. 3a (the formulation is given in Appendix A). Initially, there are three routes: Depot 1 – 2 – 1 – Depot 1, Depot 1 – 3 – Depot 1, and Depot 2 – 4 – 5 – Depot 2. The solution produced by IDMIP is given in Fig. 3b. A savings of two units is achieved by removing customer 3 from its route in Fig. 3a, moving 5 units of its demand

Table 3
Inter-depot routing algorithm.

```
S = current MDSDVRP solution, M = number of depots
maxCount = 20, cntr = 1
Repeat
  For w = 1 to M
    Let S_w be the VRP solution with depot w
    Apply ERTR to S_w
  End-For
  Apply one-point moves across all routes of all depots
  cntr = cntr + 1
Until (cntr > maxCount) or (no improvement of S)
Return S
```

Table 4
IDH heuristic for the MDSDVRP.

```
M = number of depots
Assign customers to depots
If (the number of customers N is less than 120)
  Set MIP time limit T_1 to 250/M seconds
  Set MIP time limit T_2 to 250 s
Else T_1 = 400/M, T_2 = 400
Set max neighborhood size L = 10
Solve the SDVRP for each depot and its assigned customers separately
  using EMIP-MDA + ERTR with time limit T_1, denote solution by S
S_1 = IDMIP solution on S with time limit T_2 and neighborhood size L
S_2 = solution after IDR is performed on S_1
Return S_2
```

immediately prior to customer 2, and moving 5 units of its demand immediately prior to customer 4.

3.4. Improving routes with an inter-depot routing algorithm

By solving the SDVRP for each depot and the customers assigned to it, we generate an initial solution S to the MDSDVRP. Using S, an IDMIP is formulated and solved. We point out that a small IDMIP can be time-consuming to solve (a 50-node instance can have as many as 550 integer variables, 500 continuous variables, and 1800 constraints). Therefore, we set a run-time limit that takes into account the size of the instance. We solve the IDMIP and denote the solution by S_1.

Finally, we perform a route clean-up procedure. Using S_1, we create an MDVRP instance, denoted by I, by considering each visit to each customer on a route in S_1 as a distinct customer in I whose demand is the amount serviced on that visit. For example, if there is a split delivery at customer i, with d_r units being delivered on route r and d_q units being delivery on route q ($d_r + d_q = D_i$), then in I, we create two distinct customers at the same location as i, one with demand d_r and one with demand d_q.

We apply the inter-depot routing algorithm (IDR) given in Table 3 to I using S_1 as the initial solution. IDR is strictly a routing heuristic (no new splits are created during its execution). IDR improves the routes of each depot separately using ERTR and then performs one-point moves across all routes including those of different depots. The complete description of our heuristic for the MDSDVRP (denoted by IDH) is given in Table 4.

4. Computational experiment with IDH

4.1. Analysis on modified MDVRPs

Since the MDSDVRP is a new problem, there are no benchmark instances that we could use to analyze the performance of IDH. We created new instances by modifying 10 MDVRPs originally proposed by Christofides and Eilon (1969) and Gillett and Johnson (1976). We used the node locations from these instances and changed the customer demands. We let the demand at a customer be a random integer generated uniformly in the interval $[aQ, bQ]$, where Q is the vehicle capacity and $[a, b]$ $(0 < a < b < 1)$ is the fractional demand range. We used three fractional demand ranges ([.1,.9], [.3,.7], and [.7,.9]) giving a total of $10 \times 3 = 30$ instances. Complete descriptions of these instances are given in Gulczynski (2010).

We changed the customer demands because, in the original MDVRPs, the demands are too small, relative to vehicle capacity, for split deliveries to have a significant effect on the solution. When customer demands are small, there is little advantage to splitting deliveries, so the solutions with and without split deliveries are nearly the same (Archetti, Savelsbergh, & Speranza, 2008).

We applied IDH to these 30 instances. We measured the improvements to solutions by allowing split deliveries, and the improvements by allowing inter-depot split deliveries. In Table 5, we give our computational results. In column one, we give the instance number. In columns two through four, we give the number of customers (N), the number of depots (M), and the fractional demand range. In column five, we give the solution values generated by IDH without applying the EMIP-MDA and the IDMIP (no split deliveries). In column six, we give the solution values generated by IDH without the IDMIP (no split deliveries between routes of different depots). In column seven, we give the solution values generated by IDH. In the last column, we give the run times of IDH in seconds. The integer programs in IDMIP are solved with ILOG CPLEX 10.0 and Visual C++ (version 6.0) using a 3.0 GHz Pentium 4 processor and 512 MB of RAM.

In Table 5, we see that, on these 30 instances, we reduced the distance traveled by 6.32%, on average, by allowing split deliveries between routes of the same depot. We reduced the distance traveled by an additional 1.04%, on average, by allowing split deliveries between routes of different depots. The total improvement between the IDH solutions with and without split deliveries is 7.30%, on average.

We note that the improvements achieved in the "IDH solution splits between depots" column of Table 5 are accomplished by allowing inter-depot split deliveries (splits between routes of different depots) and inter-depot node moves. The size of the improvement depends, in part, on the distances between the customers and their two closest depots. For instances in which *few* customers are located nearly equidistant from their two closest depots, there is not much need for inter-depot split deliveries and node moves, so the inter-depot routines of IDH (IDMIP and IDR) will not likely improve a solution much. For instances in which *many* customers are located nearly equidistant from their two closest depots, IDMIP and IDR can be very effective in improving a solution.

For example, on MDSD4, there is almost no improvement for the three instances (demand ranges) produced by inter-depot splits (there is a slight improvement of 0.04% for demand range [.3,.7]), because only a few customers are located nearly equidistant from their two closest depots (13 customers (13%) have a ratio $\frac{\lambda}{\lambda'} \geqslant \epsilon$, where λ is the distance between a customer and its closest depot, λ' is the distance between a customer and its second closest depot, and $\epsilon = .75$ is the tolerance defined in Section 3.1 that we used in practice). In contrast, in MDSD8, many customers are located nearly equidistant from their two closest depots (54 customers (22%) have a ratio $\frac{\lambda}{\lambda'} \geqslant \epsilon$), and the average improvement for the three instances is 1.78%.

Overall, the average improvement achieved by IDH by allowing inter-depot split deliveries is at least 1% for 19 of the 30 instances and at least 1.5% for 10 of the 30 instances. This type of reduction in travel cost would be significant for many routing managers, especially during times of increasing fuel prices and economic belt-tightening.

4.2. Performance on MDSDVRPs with visually estimated solutions

To compare the solutions of IDH against benchmark solutions, we created 12 instances (SQ1–SQ12) that have very good visually

Table 5
Computational results for IDH.

Instance	N	M	Demand range	IDH solution no splits	IDH solution splits on same depot	IDH solution splits between depots	Time[1] (s)
MDSD1	50	4	[.1,.9]	1067.36	1018.22	1018.22	634.97
			[.3,.7]	1027.65	1008.91	990.85	614.86
			[.7,.9]	1422.68	1365.75	1344.99	614.63
MDSD2	75	5	[.1,.9]	1365.26	1297.76	1289.06	687.64
			[.3,.7]	1290.35	1240.82	1223.57	681.98
			[.7,.9]	1808.49	1728.80	1705.98	680.52
MDSD3	100	2	[.1,.9]	2749.47	2636.54	2624.41	654.56
			[.3,.7]	2703.78	2604.16	2558.33	657.76
			[.7,.9]	4378.35	3919.89	3878.34	660.55
MDSD4	100	2	[.1,.9]	2514.22	2393.23	2393.23	639.72
			[.3,.7]	2507.00	2337.59	2336.65	651.33
			[.7,.9]	3922.66	3525.24	3525.24	645.45
MDSD5	100	3	[.1,.9]	2121.28	1966.67	1963.13	656.05
			[.3,.7]	1990.83	1876.73	1871.47	665.25
			[.7,.9]	3007.21	2793.81	2772.58	649.44
MDSD6	100	4	[.1,.9]	2090.91	1985.72	1963.68	657.05
			[.3,.7]	2014.99	1908.28	1887.48	689.08
			[.7,.9]	2896.41	2707.57	2696.47	664.06
MDSD7	249	2	[.1,.9]	17145.99	16376.96	16096.91	953.58
			[.3,.7]	17637.26	16410.12	16136.07	944.08
			[.7,.9]	28993.25	25988.47	25502.49	937.56
MDSD8	249	3	[.1,.9]	14114.66	13458.80	13258.26	969.70
			[.3,.7]	14675.92	13707.36	13444.18	948.76
			[.7,.9]	23629.23	21326.62	20915.02	987.36
MDSD9	249	4	[.1,.9]	12784.86	12044.79	11959.27	960.80
			[.3,.7]	13473.88	12330.74	12176.61	942.39
			[.7,.9]	21223.79	19048.59	18844.77	987.05
MDSD10	249	5	[.1,.9]	12161.69	11572.88	11377.30	938.83
			[.3,.7]	12934.74	11960.70	11831.52	980.63
			[.7,.9]	19841.86	17959.15	17777.76	961.67
Average deviation					6.32[2]	1.04[3]	

[1] 3.0 GHz Pentium 4 processor.
[2] 100[1 − (IDH Solution Splits on Same Depot/IDH Solution No Splits)]%.
[3] 100[1 − (IDH Solution Splits Between Depots/ IDH Solution Splits on Same Depot)]%.

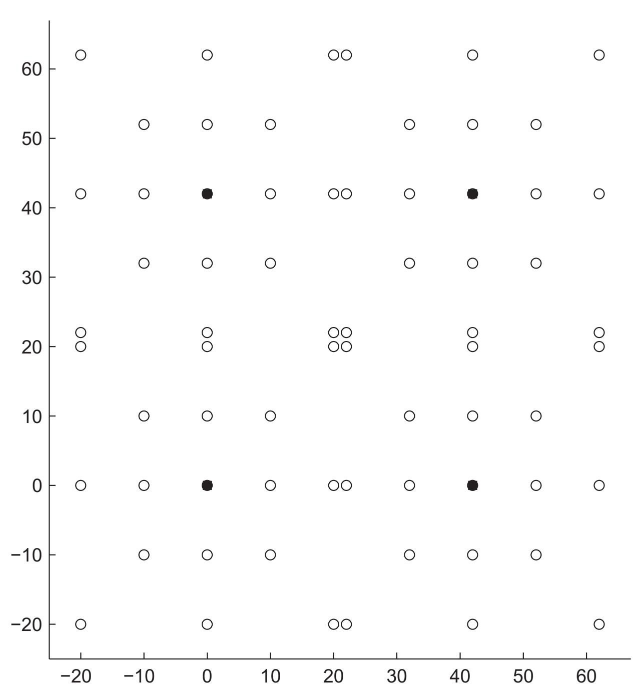


Fig. 4. SQ3 has 64 customers and four depots. The solid dots are the depots and the open dots are the customers.

estimated solutions. SQ1–SQ12 have customers located on concentric squares centered at the depots, and vary in size from 32 customers to 240 customers and from two depots to five depots. In Fig. 4, we show instance SQ3 with 64 customers and four depots. The algorithm used to generate the square instances is given in Appendix B. All instances are given in Gulczynski (2010).

In Fig. 5, we show a portion of the visually estimated solution for SQ3. There are eight customers (nodes 1–8) and two depots. Customers 1 and 7 have demand 90, customers 2 and 8 have demand 60, customer 3 has demand 85, customer 4 has demand 55, customer 5 has demand 95, and customer 6 has demand 65. The vehicle capacity is 100 units.

In Fig. 5, there are six routes. Three routes use depot 1. One vehicle starts at depot 1, delivers 80 units to customer 1, delivers

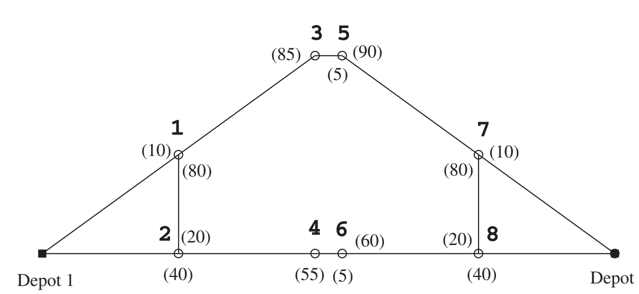


Fig. 5. A portion of a visually estimated solution for instance SQ3.

Table 6
Computational results for IDH on 12 square instances.

Instance	N	M	IDH solution no splits	IDH solution splits on same depot	IDH solution splits between depots	Estimated solution	Time[1] (s)
SQ1	32	2	1168.61	1063.08	1063.08	1057.69	638.21
SQ2	48	3	1760.58	1620.98	1601.02	1588.53	634.49
SQ3	64	4	2356.68	2151.39	2142.11	2131.37	559.59
SQ4	80	5	2945.88	2699.63	2684.02	2662.21	604.27
SQ5	64	2	3872.47	3444.71	3434.71	3422.19	646.75
SQ6	96	3	5816.36	5206.26	5142.06	5135.29	653.13
SQ7	128	4	7764.32	6931.80	6869.14	6860.39	924.82
SQ8	160	5	9705.41	8630.72	8600.60	8573.48	928.18
SQ9	96	2	8189.25	7231.32	7109.71	7050.62	696.69
SQ10	144	3	12189.77	10784.16	10586.51	10577.93	939.77
SQ11	192	4	16262.29	14393.48	14135.80	14117.24	947.85
SQ12	240	5	20327.92	18199.20	17739.64	17644.55	940.91
Average deviation				10.21[2]	1.07[3]	0.54[4]	

[1] 3.0 GHz Pentium 4 processor.
[2] 100[1 − (IDH Solution Splits on Same Depot/IDH Solution No Splits)]%.
[3] 100[1 − (IDH Solution Splits Between Depots/IDH Solution Splits on Same Depot)]%.
[4] 100[1 − (Estimated Solution/IDH Solution)]%.

20 units to customer 2, and returns to depot 1. A second vehicle starts at depot 1, delivers 10 units to customer 1, delivers 85 units to customer 3, delivers 5 units to customer 5, and returns to depot 1. (For visual simplicity, we assume the vehicle travels back through customers 3 and 1 on its path from customer 5 to depot 1. The added distance from this assumption is very small. Additional details are given in Appendix C.) A third vehicle starts at depot 1, delivers 40 units to customer 2, delivers 55 units to customer 4, delivers 5 units to customer 6, and returns to depot 1. Three routes use depot 2. One vehicle starts at depot 2, delivers 80 units to customer 7, delivers 20 units to customer 8, and returns to depot 2. A second vehicle starts at depot 2, delivers 10 units to customer 7, delivers 90 units to customer 5, and returns to depot 2. A third vehicle starts at depot 2, delivers 40 units to customer 8, delivers 60 units to customer 6, and returns to depot 2.

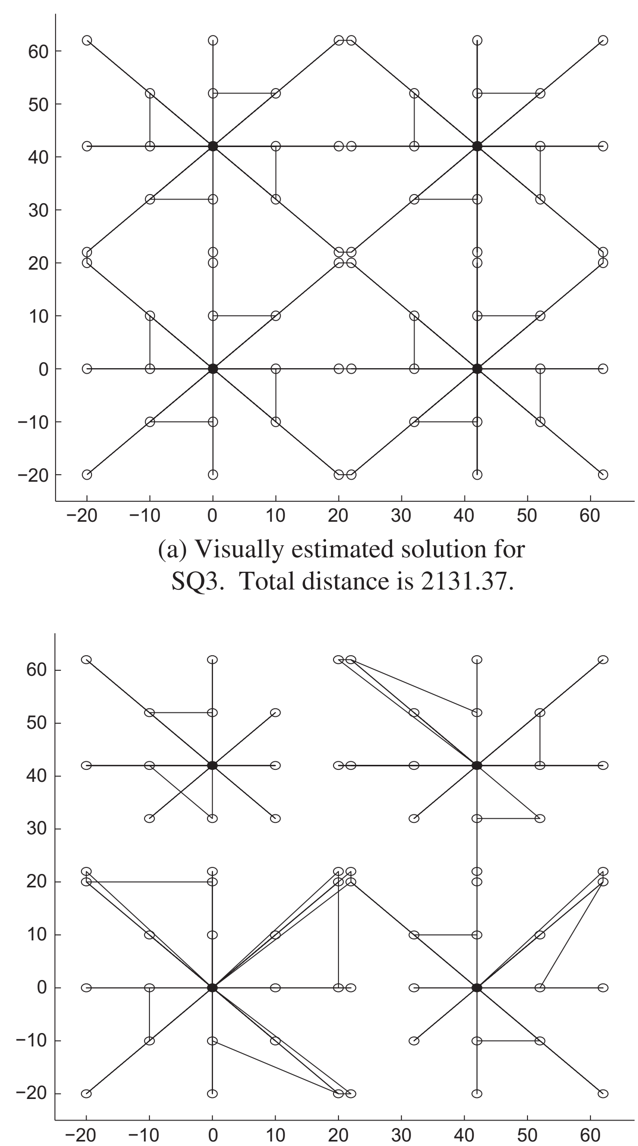
In Fig. 5, the deliveries to customers 1 and 2 are split between vehicles using depot 1. The deliveries to customers 7 and 8 are split between vehicles using depot 2. The deliveries to customers 5 and 6 are split between a vehicle using depot 1 and a vehicle using depot 2. The routes of the visually estimated solutions for all square instances follow this basic structure.

In Table 6, we see that, on the 12 square instances, the distance traveled is reduced by 10.21%, on average, by allowing split deliveries between routes of the same depot. The distance traveled is reduced by an additional 1.07%, on average, by allowing split deliveries between routes of different depots. IDH solutions with split deliveries are 11.65% better, on average, than IDH solutions without split deliveries.

IDH performs very well on the square instances. On average, the estimated solution is only 0.54% below the solution produced by IDH. This suggests that IDH is a very high-quality heuristic for the MDSDVRP.

In Fig. 6, we show the visually estimated solution and the solution produced by IDH for instance SQ3. The visually estimated solution has 48 routes, 42 total split deliveries, and 10 split deliveries among vehicles using different depots with routes that have a structure like those in Fig. 5. The IDH solution has 51 routes, 25 total split deliveries, and 2 splits between vehicles using different depots. The IDH solution for SQ3 is 0.50% larger than the estimated solution.

We do not know whether or not the visually estimated solutions are optimal, but they look to be close to optimal. They have two features that are usually desirable in vehicle routing solutions. All vehicles are filled to capacity, and the insertion costs of most customers are very small or zero. That is, the service to most customers on a route adds little or no additional distance to the route.



(a) Visually estimated solution for SQ3. Total distance is 2131.37.

(b) IDH solution for SQ3. Total distance is 2142.11.

Fig. 6. Visually estimated solution and IDH solution for instance SQ3.

Table 7
Generator for square instances.

```
(Get depot coordinates)
For m = 1 to M
  p = ⌊√m⌋, q = p^2
  If (m − q ⩽ p), a = p, b = m − q
  Else a = (p + 1)^2 − m − 1, b = p
  u_m = (20G + 2)a, v_m = (20G + 2)b
End-For
(Get customer coordinates and demands)
i = 1
For m = 1 to M
  For g = 1 to G
    For j = −1 to 1
      For k = −1 to 1
        If (j ≠ 0) or (k ≠ 0)
          x_i = 10gj + u_m, y_i = 10gk + v_m
          If (|j| − |k| = 0), D_i = 90
          Else D_i = 60
          If (g = G)
            If (j = −1) and (φ_m^− = 1) and (m mod 2 = 0)
              D_i = D_i − 5
            Else If (j = −1) and (φ_m^− = 1) and (m mod 2 = 1)
              D_i = D_i + 5
            Else If (j = 1) and (φ_m^+ = 1) and (m mod 2 = 0)
              D_i = D_i − 5
            Else If (j = 1) and (φ_m^+ = 1) and (m mod 2 = 1)
              D_i = D_i + 5
            Else If (k = −1) and (ω_m^− = 1) and (m mod 2 = 0)
              D_i = D_i − 5
            Else If (k = −1) and (ω_m^− = 1) and (m mod 2 = 1)
              D_i = D_i + 5
            Else If (k = 1) and (ω_m^+ = 1) and (m mod 2 = 0)
              D_i = D_i − 5
            Else If (k = 1) and (ω_m^+ = 1) and (m mod 2 = 1)
              D_i = D_i + 5
          End-If
          i = i + 1
        End-If
      End-For (k loop)
    End-For (j loop)
  End-For (g loop)
End-For (m loop)
```

For example, in Fig. 5, a vehicle visits customers 2, 4, and 6. The insertion costs of both customers 2 and 4 on this route are zero, and the insertion cost of customer 6 is very small. Since this is the case for most customers in the visually estimated solutions, it appears that each route is very efficient and the solutions are very high quality. A detailed description of the estimated solutions is given in Appendix C. All solutions are given in Gulczynski (2010).

5. Conclusions

In this paper, we described the multi-depot split delivery vehicle routing problem which is a new problem in the VRP literature.

Table 8
Specifications of the square instances.

Instance	N	M	G
SQ1	32	2	2
SQ2	48	3	2
SQ3	64	4	2
SQ4	80	5	2
SQ5	64	2	4
SQ6	96	3	4
SQ7	128	4	4
SQ8	160	5	4
SQ9	96	2	6
SQ10	144	3	6
SQ11	192	4	6
SQ12	240	5	6

We developed a heuristic for the MDSDVRP that combined an algorithm for the traditional SDVRP with an inter-depot mixed integer program and an inter-depot routing algorithm. We applied our heuristic to 30 modified MDVRPs and measured the improvements achieved by splitting deliveries between routes of the same depot and splitting deliveries between routes of different depots.

We generated 12 new test instances that have high-quality, visually estimated solutions. Our heuristic produced very good solutions to these 12 instances. Overall, our solution procedure was very effective in solving a wide range of instances.

In future work, we could consider imposing minimum delivery amounts on vehicles when splitting the service to a customer and having time windows for visits to customers. Time windows are particularly relevant when modeling problems encountered in the waste collection industry.

Acknowledgment

We thank Chris Groër for providing his source code. The third author's research was supported in part by a Kogod Research Professorship at American University.

Appendix A

We present the IDMIP formulation using the example given in Fig. 3a as the initial solution. The three routes of the initial solution are (1) 0-2-1-0, (2) 0-3-0, and (3) 6-4-5-6, where nodes 0 and 6 represent depot 1 and depot 2, respectively. In this example, for simplicity, we omit the decision variables for customers 1 and 5. These variables are zero in the optimal solution. The IDMIP formulation is as follows.

$$
\begin{aligned}
\text{maximize}\quad & b_2(c_{02}+c_{21}-c_{01})+2b_3c_{03}+b_4(c_{64}+c_{45}-c_{65})\\
& -m_{23}(c_{02}+c_{23}-c_{03})-m_{24}(c_{62}+c_{24}-c_{64})\\
& -m_{32}(c_{03}+c_{32}-c_{02})-m_{34}(c_{63}+c_{34}-c_{64})\\
& -m_{42}(c_{04}+c_{42}-c_{02})-m_{43}(c_{04}+c_{43}-c_{03})\\
\text{subject to}\quad & d_{32}+d_{42}-d_{23}-d_{24}\leqslant Q_1\\
& d_{23}+d_{43}-d_{32}-d_{34}\leqslant Q_2\\
& d_{24}+d_{34}-d_{42}-d_{43}\leqslant Q_3\\
& d_{23}+d_{24}\leqslant D_2\\
& d_{32}+d_{34}\leqslant D_3\\
& d_{42}+d_{43}\leqslant D_4\\
& d_{23}+d_{24}\geqslant D_2b_2\\
& d_{32}+d_{34}\geqslant D_3b_3\\
& d_{42}+d_{43}\geqslant D_4b_4\\
& D_2m_{23}\geqslant d_{23}\\
& D_2m_{24}\geqslant d_{24}\\
& D_3m_{32}\geqslant d_{32}\\
& D_3m_{34}\geqslant d_{34}\\
& D_4m_{42}\geqslant d_{42}\\
& D_4m_{43}\geqslant d_{43}\\
& 1-b_2\geqslant m_{32}+m_{42}\\
& 1-b_3\geqslant m_{23}+m_{43}\\
& 1-b_4\geqslant m_{24}+m_{34}\\
& d_{ij}\geqslant 0\quad \text{for } i,j=2,3,4\\
& b_i=0,1\quad \text{for } i=2,3,4\\
& m_{ij}=0,1\quad \text{for } i,j=2,3,4
\end{aligned}
$$

In this example, we have $Q_1 = 5$, $Q_2 = 15$, $Q_3 = 5$, and $D_2 = D_3 = D_4 = 10$. The symmetric distances are given by $c_{02} = 3$, $c_{03} = 2$, $c_{04} = 4$, $c_{23} = 2$,

$c_{24} = 3, c_{26} = 4, c_{34} = c_{36} = 2$, and $c_{46} = 3$. The objective function is given by

$$\text{maximize} \quad 3b_2 + 4b_3 + 3b_4 - 3m_{23} - 4m_{24} - m_{32} - m_{34} - 4m_{43} - 4m_{42}$$

Since the decision variables for customers 1 and 5 are omitted in this example, constraints (7) and (8) are not applicable, so they are not presented. The objective function is maximized when $b_3 = 1$, $m_{32} = 1$, $m_{34} = 1$, $d_{32} = 5$, $d_{34} = 5$, and all other decision variables are 0. A maximum savings of two units is produced by removing customer 3 from its route, reallocating 5 units of its demand before customer 2, and reallocating 5 units of its demand before customer 4. This solution is given in Fig. 3b.

Appendix B

We present the generator for the 12 square instances. In Table 7, we give the pseudo-code for the generation algorithm. The parameters of the algorithm are defined as follows. Let N be the number of customers and M be the number of depots. Let G be the number of concentric squares of customers centered at a depot. There are eight customers on each square, so $N = 8MG$. Let (u_m, v_m) be the coordinates of depot m and (x_i, y_i) be the coordinates of customer i. D_i is the demand of customer i. The vehicle capacity is 100 units.

ϕ_m^-, ϕ_m^+, ω_m^-, ω_m^+ are defined as follows. Given depot m with coordinates (u_m, v_m), if there is a depot located at (x, v_m), where $x < u_m$, then $\phi_m^- = 1$, otherwise $\phi_m^- = 0$. If there is a depot located at (x, v_m), where $x > u_m$, then $\phi_m^+ = 1$, otherwise $\phi_m^+ = 0$. If there is a depot located at (u_m, y), where $y < v_m$, then $\omega_m^- = 1$, otherwise $\omega_m^- = 0$. If there is a depot located at (u_m, y), where $y > v_m$, then $\omega_m^+ = 1$, otherwise $\omega_m^+ = 0$.

In Table 8, we give the number of customers (N), the number of depots (M), and the number of concentric squares (G) centered at each depot for the 12 square instances.

Appendix C

We describe the visually estimated solutions to the square instances. Each estimated solution is composed of wedge-shaped patterns of routes. Each wedge is one quarter of the routes for a depot (i.e., there are four wedges per depot). There are three different types of wedges: 4-customer, 8-customer, and 12-customer. In Fig. 7a, Fig. 7b, and Fig. 7c, we show the routes of the 4-customer, 8-customer, and 12-customer wedges, respectively. Node 0 is the depot. The odd-numbered customers have demand 90, and the even-numbered customers all have demand 60.

In Table 9, we give the routes of the wedges in Fig. 7. In each row, we give the route number, the customer sequence and delivery amounts on the route, and the distance traveled. For example, the first route of the 4-customer wedge leaves the depot, delivers 10 units to customer 1, delivers 90 units to customer 3, and returns to the depot. The distance traveled on the route is 56.57 units. We also give the total distance traveled in the wedges.

The visually estimated solution to each square instance is composed of various orientations of the wedges shown in Fig. 7. We generate inter-depot split deliveries in the estimated solutions by changing the demands of customers close to the border of two wedges. Specifically, for wedges of adjacent depots (call them depot 1 and depot 2), we subtract five units of demand from the outermost customers of depot 1's wedge and add five units of demand to the outermost customers of depot 2's wedge. We then extend two routes of depot 1 to deliver five units to the outermost customers of depot 2's wedge. This is illustrated in Fig. 5. We see two adjacent 4-customer wedges. We changed the demands of the boundary customers of the wedges (customers 3–6). Customer 3 has demand 85 (90 − 5), customer 4 has demand 55 (60 − 5), customer 5 has demand 95 (90 + 5), and customer 6 has demand 65 (60 + 5). Two routes of depot 1's wedge have been extended to visit customers 5 and 6, delivering five units to each of them. The distance added by extending a route is four units. The boundary customers of adjacent wedges (e.g., customers 3 and 5) are two units apart, so we add two units traveling to a new customer on a route and two units traveling back.

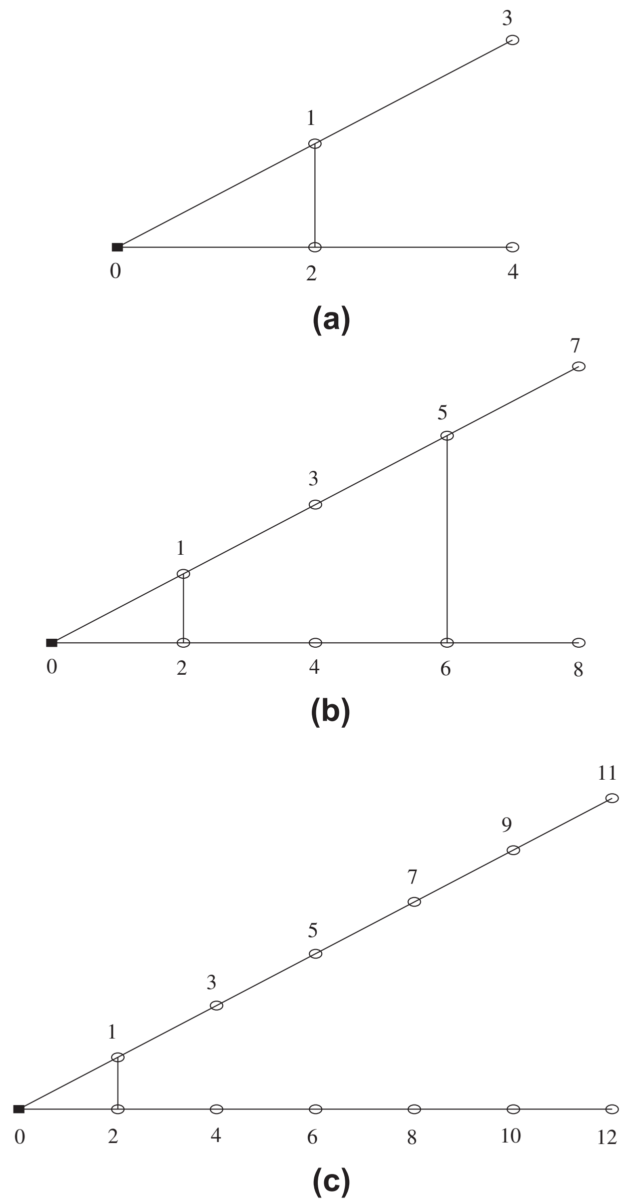


Fig. 7. Route wedges of the square instances.

In Fig. 6a, we show all the routes of the visually estimated solution to instance SQ3. There are four depots and 16 total 4-customer wedges (four wedges per depot). There are 10 inter-depot splits. The distance traveled on each 4-customer wedge is 130.71 (Table 9), and each inter-depot split adds four units, so the total distance of the estimated solution is $16 \times 130.71 + 4 \times 10 = 2131.37$.

In Table 10, we give the specifications of the visually estimated solution to each square instance. We give the instance, the number of depots (M), the type of wedge used in the solution, the number of inter-depot splits (L) in the solution, and the total distance traveled of the solution. The general formula for distance traveled is $4 \times M \times W + 4 \times L$, where W is the distance traveled of the wedge given in Table 9.

Table 9
Route wedges of the estimated solutions.

Route number	Customer sequence and delivery amounts	Distance traveled
4-Customer wedge		
1	0 1(10) 3(90) 0	56.57
2	0 1(80) 2(20) 0	34.14
3	0 2(40) 4(60) 0	40.00
Total distance traveled		130.71
8-Customer wedge		
1	0 1(10) 3(90) 0	56.57
2	0 5(10) 7(90) 0	113.14
3	0 1(80) 2(20) 0	34.14
4	0 5(80) 6(20) 0	102.43
5	0 2(40) 4(60) 0	40.00
6	0 6(40) 8(60) 0	80.00
Total distance traveled		426.28
12-Customer wedge		
1	0 1(50) 3(50) 0	56.57
2	0 3(40) 5(60) 0	84.85
3	0 5(30) 7(70) 0	113.14
4	0 7(20) 9(80) 0	141.42
5	0 9(10) 11(90) 0	169.71
6	0 1(40) 2(60) 0	34.14
7	0 4(60) 6(40) 0	60.00
8	0 6(20) 8(60) 10(20) 0	100.00
9	0 10(40) 12(60) 0	120.00
Total distance traveled		879.83

Table 10
Specifications of the estimated solutions to the square instances.

Instance	M	Wedge type	L	Distance traveled
SQ1	2	4-Customer	3	1057.69
SQ2	3	4-Customer	5	1588.53
SQ3	4	4-Customer	10	2131.37
SQ4	5	4-Customer	12	2662.21
SQ5	2	8-Customer	3	3422.19
SQ6	3	8-Customer	5	5135.29
SQ7	4	8-Customer	10	6860.39
SQ8	5	8-Customer	12	8573.48
SQ9	2	12-Customer	3	7050.62
SQ10	3	12-Customer	5	10577.93
SQ11	4	12-Customer	10	14117.24
SQ12	5	12-Customer	12	17644.55

We point out that we can find slightly better solutions than those given in Table 10 by not "backtracking" on some routes. For example, in Fig. 5, we see a vehicle based at depot 1 that visits customers 1, 3, and 5. In the visually estimated solution this vehicle travels back through customers 3 and 1 when returning to the depot from customer 5. If, instead, the vehicle were to travel from customer 5 directly back to the depot, the total distance of this route would decrease by a small amount (about 0.07% on average over all instances). However, if we were to eliminate backtracking in the estimated solutions, the visual presentation of the routes would be cluttered and the formula for calculating their distances would be complicated.

References

Afshar A. (2011). A mathematical framework for optimizing disaster relief logistics. Ph.D. dissertation. Department of Civil and Environmental Engineering, University of Maryland, College Park.

Angelelli, E., & Speranza, M. G. (2002). The application of a vehicle routing model to a waste-collection problem: Two case studies. *Journal of the Operational Research Society, 53*, 944–952.

Archetti, C., Savelsbergh, M., & Speranza, M. G. (2008). To split or not to split: That is the question. *Transportation Research Part E, 44*, 114–123.

Archetti, C., & Speranza, M. G. (2004). Vehicle routing in the 1-skip collection problem. *Journal of the Operational Research Society, 55*, 717–727.

Archetti, C., & Speranza, M. G. (2008). The split delivery vehicle routing problem: A survey. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges* (pp. 103–122). New York: Springer.

Archetti, C., Speranza, M. G., & Hertz, A. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science, 40*, 64–73.

Chao, Y.-L. (2008). Time series analysis of the effects of refuse collection on recycling: Taiwan's 'keep trash off the ground' measure. *Waste Management, 28*, 859–869.

Chao, I-M., Golden, B., & Wasil, E. (1993). A new heuristics for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences, 13*, 371–406.

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

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly, 20*, 309–318.

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks, 30*, 105–119.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2004). Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society, 55*, 542–546.

Crevier, B., Cordeau, J.-F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research, 176*, 756–773.

De Franceschi, R., Fischetti, M., & Toth, P. (2006). A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming, 105*, 471–499.

Derigs, U., Li, B., & Vogel, U. (2009). Local search-based metaheuristics for the split delivery vehicle routing problem. *Journal of the Operational Research Society, 61*, 1356–1364.

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

Dror, M., & Trudeau, P. (1990). Split delivery routing. *Naval Research Logistics, 37*, 383–402.

Gillett, B. E., & Johnson, J. G. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega, 4*, 711–718.

Giosa, I. D., Tansini, I. L., & Viera, I. O. (2002). New assignment algorithms for the multi-depot vehicle routing problem. *Journal of the Operational Research Society, 53*, 977–984.

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks, 7*, 113–148.

Golden, B., & Wasil, E. (1987). Computerized vehicle routing in the soft drink industry. *Operations Research, 35*, 6–17.

Groër, C., Golden, B., & Wasil, E. (2010). A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation, 2*, 79–101.

Gulczynski, D. (2010). *Integer programming-based heuristics for vehicle routing problems*. Ph.D. dissertation. Department of Mathematics, University of Maryland, College Park.

Gulczynski, D., Golden, B., & Wasil, E. (2008). Recent developments in modeling and solving the split delivery vehicle routing problem. In Z. Chen & S. Raghavan (Eds.), *Tutorial in operations research 2008: State-of-the-art decision making tools in the information-intensive age* (pp. 170–180). Hanover, MD: INFORMS.

Gulczynski, D., Golden, B., & Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Science Part E, 46*, 612–626.

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

Li, J., Chu, F., & Chen, H. (2011). Coordination of split deliveries in one-warehouse multi-retailer distribution systems. *Computers & Industrial Engineering, 60*, 291–301.

Li, F., Golden, B., & Wasil, E. (2005). Very large-scale vehicle routing: New test problems, algorithms, and results. *Computers & Operations Research, 32*, 1165–1179.

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

Mota, E., Campos, V., & Corberán, Á. (2007). A new metaheuristic for the vehicle routing problem with split deliveries. In C. Cotta & J. van Hemert (Eds.), *Lecture notes in computer science. Evolutionary computation in combinatorial optimization* (Vol. 4446, pp. 121–129). Berlin: Springer.

Pisinger, D., & Røpke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*, 2403–2435.

Polacek, M., Hartl, R. F., Doerner, K., & Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics, 10*, 613–627.

Raft, O. M. (1982). A modular algorithm for an extended vehicle scheduling problem. *European Journal of Operational Research, 11*, 67–76.

Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research, 23*, 229–235.

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society, 50*, 1034–1042.

Thangiah, S. R., & Salhi, S. (2001). Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem. *Applied Artificial Intelligence, 15*, 361–383.

Tillman, F. A., & Cain, T. M. (1972). An upperbound algorithm for the single and multiple terminal delivery problem. *Management Science, 18*, 664–682.
Wang, W., Fung, R., & Chai, Y. (2004). Approach of just-in-time distribution requirements planning for supply chain management. *International Journal of Production Economics, 91*, 101–107.
Wren, A., & Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly, 23*, 333–344.
Yellow, P. C. (1970). A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly, 21*, 281–283.