The Consistent Vehicle Routing Problem

Chris Groër
Applied Mathematics and Scientific Computation Program, University of Maryland, College Park, Maryland 20742, cgroer@gmail.com

Bruce Golden
R. 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, DC 20016, ewasil@american.edu

In the small package shipping industry (as in other industries), companies try to differentiate themselves by providing high levels of customer service. This can be accomplished in several ways, including online tracking of packages, ensuring on-time delivery, and offering residential pickups. Some companies want their drivers to develop relationships with customers on a route and have the same drivers visit the same customers at roughly the same time on each day that the customers need service. These service requirements, together with traditional constraints on vehicle capacity and route length, define a variant of the classical capacitated vehicle routing problem, which we call the consistent VRP (ConVRP). In this paper, we formulate the problem as a mixed-integer program and develop an algorithm to solve the ConVRP that is based on the record-to-record travel algorithm. We compare the performance of our algorithm to the optimal mixed-integer program solutions for a set of small problems and then apply our algorithm to five simulated data sets with 1,000 customers and a real-world data set with more than 3,700 customers. We provide a technique for generating ConVRP benchmark problems from vehicle routing problem instances given in the literature and provide our solutions to these instances. The solutions produced by our algorithm on all problems do a very good job of meeting customer service objectives with routes that have a low total travel time.

Key words: vehicle routing; customer service; logistics

History: Received: February 1, 2008; accepted: September 14, 2008. Published online in Articles in Advance December 4, 2008.

1. Introduction
The traditional vehicle routing problem (VRP) has been studied by researchers and practitioners for nearly 50 years, dating back to the early work of Dantzig and Ramser (1959). For the most part, the focus has been on developing a set of routes for a homogeneous fleet of vehicles that minimizes the total cost or the total distance traveled by the fleet, subject to a set of constraints.

Over the last five years or so, there has been a shift in practice from fleet-focused considerations to those that are customer focused. Drivers for United Parcel Service (UPS) “…form a real bond with customers [and] take tremendous ownership of their customers and routes” (UPS 2007). This is a significant competitive advantage for UPS, as drivers have gathered sales leads that have generated an additional “…volume of more than 60 million packages a year.” Each day 103,500 UPS drivers visit 7.9 million customers and handle an average of 15.6 million packages (UPS 2007). In recent years, firms in the small package shipping industry acknowledge that it is important for customers to receive service from the same service provider at about the same time over multiple days, and that this is a key component of high-quality customer service.

We combine the service provider consistency requirements (the same driver visits the same customers at roughly the same time on each day that these customers need service) with traditional constraints on vehicle capacity and route length and define a variant of the traditional VRP that we call the consistent VRP (ConVRP).

In Table 1, we show the modeling focus for traditional VRP variants, including the ConVRP. In the past, researchers and practitioners considered the
Table 1  Modeling Focus for the VRP

<table>
<thead>
<tr>
<th>Problem</th>
<th>Objective Focus</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional VRP</td>
<td>Minimize cost or distance</td>
<td>Fleet</td>
</tr>
<tr>
<td>VRP with balance</td>
<td>Balance routes on each day</td>
<td>Driver</td>
</tr>
<tr>
<td>Period VRP</td>
<td>Account for time-sensitive demands</td>
<td>Demand</td>
</tr>
<tr>
<td>ConVRP</td>
<td>Be consistent from one day to the next with customers</td>
<td>Customer</td>
</tr>
</tbody>
</table>

Groër, Golden, and Wasil: The Consistent Vehicle Routing Problem

fleets, the driver, and the demand in their models. Customer service considerations are now being added to variants of the VRP. Campbell and Thomas (2008) present a number of industry statistics and discuss the importance of both on-time delivery and consistent service within the highly competitive package delivery industry. The ConVRP is the first VRP variant that we have come across that has a primary focus of customer satisfaction. For small package shipping companies, providing consistent service (with respect to time) can be more important in practice than saving an incremental one, two, or three percent in travel costs.

The rest of this paper is organized as follows. In §2, we model the ConVRP as a mixed-integer program (MIP). In §3, we develop an algorithm for solving the ConVRP. In §4, we report computational results on test problems. In §5, we give our conclusions.

2. Modeling the ConVRP

In the standard version of the VRP, a homogeneous fleet of vehicles is based at a single depot. Each vehicle has a fixed capacity and must leave from and return to the same depot. There are N customers, and each customer has a known demand and is serviced by exactly one visit of a single vehicle. A route must be developed for each vehicle so that all customers are serviced and the total distance traveled by the fleet is minimized.

In the ConVRP, there are D days of service requirements. Each customer must be serviced on a specific day (the days are known in advance), and each customer can receive service at most once on any day from any one of at most K identical vehicles. When a customer receives service, the same driver visits the customer at roughly the same time over the D-day planning horizon, so that the maximum arrival time variation (between latest and earliest arrival times) is no more than L time units. The service time at customer i on day d is denoted by s_{id}, and the demand at customer i on day d is denoted by q_{id}. We define t_{ij} to be the deterministic, symmetric travel time between any two locations i and j. On each day, a vehicle has a capacity of Q units and can operate for no more than T units of time. The objective is to develop a set of routes for the fleet that minimizes the total vehicle operating time over D days.

We point out that the fleet is homogeneous: all K vehicles have the same capacity and are capable
of servicing any customer. This matches the practice of many small package shipping companies that use a standard-size vehicle to provide deliveries (e.g., the UPS Big Brown Truck™). In addition, we do not account for hard time windows. In practice, most residential customers do not have hard time windows for delivery. In the ConVRP, by adhering to consistency requirements, we are able to guarantee that a customer will be serviced at roughly the same time over the next $D$ days.

We formulate the ConVRP as an MIP, and we are able to use this formulation to solve several small problems. Let $a_{id}$ equal the arrival time at customer $i$ on day $d$ and equal 0 if no service for customer $i$ is required on day $d$ ($i = 0$ is the depot). Set $w_{id} = 1$ if customer $i$ requires service on day $d$ and $w_{id} = 0$ otherwise. The decision variable $x_{ijkd}$ equals 1 if vehicle $k$ visits customer $j$ immediately after customer $i$ on day $d$ and equals 0 otherwise. The decision variable $y_{ikd}$ equals 1 if customer $i$ is visited by vehicle $k$ on day $d$ and equals 0 otherwise. Using this notation, the objective function and constraints of the MIP are as follows:

\[
\text{Minimize} \quad \sum_{d=1}^{D} \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} t_{ij} x_{ijkd}, \quad (1)
\]

s.t.

\[
y_{ikd} = 1 \quad \text{for all } k, d, \quad (2)
\]

\[
a_{id} = 0 \quad \text{for all } d, \quad (3)
\]

\[
\sum_{k=1}^{K} y_{ikd} = w_{id} \quad \text{for } i \geq 1 \text{ and all } d, \quad (4)
\]

\[
\sum_{i=1}^{N} q_{id} y_{ikd} \leq Q \quad \text{for all } k, d, \quad (5)
\]

\[
\sum_{i=0}^{N} x_{ijkd} = \sum_{i=0}^{N} x_{ijkd} = y_{ikd} \quad \text{for all } j, k, d, \quad (6)
\]

\[
w_{id} + w_{id} - 2 \leq y_{ikd} - y_{ikd} \leq - (w_{id} + w_{id} - 2)
\]

\[
\text{for all days } a_{id} \text{ and } d_{\beta}, \quad \alpha \neq \beta, \quad (7)
\]

\[
a_{id} + x_{ijkd}(s_{id} + t_{ij}) - (1 - x_{ijkd}) T \leq a_{jd}
\]

\[
\text{for all } d, k; i \geq 0, j \geq 1, \quad (8)
\]

\[
a_{id} + x_{ijkd}(s_{id} + t_{ij}) + (1 - x_{ijkd}) T \geq a_{jd}
\]

\[
\text{for all } d, k; i \geq 0, j \geq 1, \quad (9)
\]

\[
\sum_{i \in S, j \in S} x_{ijkd} \leq |S| - 1 \quad \text{for } S \subseteq \{1, 2, \ldots, N\}, \quad (10)
\]

\[
\text{with } 2 \leq |S| \leq N \text{ for all } k, d, \quad (10)
\]

\[
0 \leq a_{id} + w_{id}(s_{id} + t_{io}) - T \leq |d|
\]

\[
\text{for all } i \geq 1, \quad (11)
\]

\[
-L + T(w_{ida} + w_{ida} - 2)
\]

\[
\leq a_{id} - a_{id} \leq L - T(w_{ida} + w_{ida} - 2)
\]

\[
\text{for all } i \text{ and days } a \text{ and } d_{\beta}, \alpha \neq \beta, \quad (12)
\]

\[
x_{ijkd} \in \{0, 1\}; \quad y_{ikd} \in \{0, 1\}; \quad a_{id} \geq 0
\]

\[
\text{for all } i, j, k, d. \quad (13)
\]

The objective function (1) minimizes the total travel time of the vehicles over all days. In (2) and (3), we require that the depot be visited at time 0 by all vehicles on all days. Constraint (4) ensures that customers are visited exactly once when they require service, and (5) guarantees that each vehicle carries no more than $Q$ units on any given day. In (6), each customer has only one predecessor and one successor, and (7) ensures that each customer is served by the same driver whenever he or she requires service. Note that the quantity $(w_{ida} + w_{ida} - 2)$ is negative unless customer $i$ requires service on both days, in which case this quantity is 0. Constraints (8) and (9) determine the arrival times at the individual customers, and (8) also serves to eliminate subtours in the individual daily routes. Although they are redundant in this case, we also include the usual subtour elimination constraints (10). In our computational experiments with small problems, we found that including these constraints allowed us to halve our computation times. We note that constraint (9) could be removed if we wanted to allow the driver to wait at a location before traveling to the next customer, to provide consistent arrival times. The vehicle travel time limit is defined in (11). In (12), the difference between arrival times at customer $i$ on any two days $\alpha$ and $\beta$ is no more than $L$ units. As in (7), constraint (12) is redundant when customer $i$ does not require service on both days.

3. Solving Large ConVRPs

We now describe an algorithm for solving large-scale ConVRPs containing up to several thousand customer
locations (solving our MIP is only practical for very small problems with a dozen or so customer locations). Many of the most successful metaheuristic algorithms for solving the standard VRP are quite complicated and involve many parameters. As the ConVRP adds several complicating constraints to the VRP, we wanted to develop an algorithm with a simple structure and relatively few parameters. The main feature of our procedure is a precedence principle: if customers \( a \) and \( b \) are both served by the same vehicle on a specific day and \( a \) is serviced before \( b \), then customer \( a \) must receive service before customer \( b \) from the same vehicle on all days that they both require service. The hope is that routes created by following this precedence principle would tend to adhere to the consistency constraints. In other words, if we consider only those customers who require service on more than one day and ensure that these customers are visited in the same order, then the resulting routes should lead to consistent service, even after we introduce additional customers who require service on only one of the \( D \) days. A key goal of this paper is to study the performance of the precedence principle as a relatively simple heuristic approach to the problem of creating consistent routes.

Our algorithm has two stages. In the first stage, we generate a set of template routes by considering only those customers who require service on multiple days. In the second stage, using the template routes, we create routes for all days by removing from the template customers who do not require service on any day \( d \) and inserting customers who require service on only day \( d \). This procedure guarantees both that customers are always visited by the same vehicle when they require service and that the order of the customer visits will adhere to the precedence principle. Our hope is that the customers will also receive service at roughly the same time each day that they are serviced.

The template routes provide the primary structure of the routes for each of the \( D \) days, and our procedure applies local search in an attempt to improve these routes. However, because the template routes themselves are never actually traversed by the vehicles, it is unclear how to treat the restrictions on a vehicle’s total travel time and capacity. Thus, the actual travel time required to traverse a template route may be substantially larger than the time limit \( T \), and the load allowed on a template route may be larger than the actual allowed capacity \( Q \). To address these issues, when we are constructing and improving the template routes, we periodically derive each daily route from the template and check its feasibility with respect to travel time and capacity. If the total travel time of any daily route exceeds \( T \), or if a particular vehicle’s load exceeds \( Q \), then we decrease the bound on the length or capacity of a template route and regenerate the template routes until the daily route becomes feasible. In contrast, if the template routes lead to daily routes that have travel times that are substantially smaller than \( T \) or capacities that are smaller than \( Q \), then we increase the relevant limit and regenerate the template routes. This allows the daily routes to either have longer total travel times or greater vehicle loads.

Our algorithm is based on the record-to-record (RTR) travel algorithm used by Li et al. (2005) to solve very large-scale VRPs and is denoted by ConRTR (ConVRP RTR travel). We improve solutions using the operations specified by Li et al. (2005): one-point move, two-point move, and two-opt move. In a one-point move, we try to move each customer in the existing solution to a new position on the same route or on a different route. In a two-point move, we try to exchange the positions of two customers. We try the usual two-opt moves within a route and between routes by replacing two existing edges with two new edges. These improvement moves are shown in Figure 1.

The ConRTR algorithm attempts to create a high-quality solution to the ConVRP by creating a set of template routes that can be used to construct feasible routes for each of the days through simple removal and insertion procedures. The template is improved by repeatedly applying these three local search operators in a diversification phase followed by an improvement phase. In the diversification phase, we try to explore new areas of the solution space by accepting both improving and deteriorating moves. In the improvement phase, we attempt to improve the current solution as much as possible by accepting only improving moves until we reach a local minimum.

**Step 1. Initialization.**

1(a). We are given \( N \) total customers, a fleet of vehicles each with maximum total travel time \( T \)
Figure 1 Improvement Operators Used in ConRTR

(a) One-point move

(b) Two-point move

(c) Two-opt move within a single route

(d) Two-opt move within two routes

and capacity \( Q \), and \( D \) days of service requirements. \( C \) denotes the current set of template routes being considered, \( F \) denotes the most recently generated set of template routes that is known to lead to feasible routes for each day, and \( F^* \) represents the set of template routes that leads to the lowest total travel time for the \( D \) days. The value of \( I \) is the number of iterations in the diversification phase, \( J \) is the maximum number of nonimproving iterations allowed before returning a solution, \( \alpha \) represents a tolerance for the amount of deterioration allowed in the local search, and \( \lambda \) is a parameter (Yellow 1970) used in the Clarke and Wright algorithm to quickly generate multiple initial solutions. Given a set of template routes \( S \), let \( f(S) \) represent the total travel time of all \( D \) routes if they are feasible.

1(b). Set \( I = 30 \), \( J = 5 \), \( \alpha = 0.01 \), \( l = 1 \), and \( \lambda \in [0.6, 1.0, 1.4] \).

1(c). Set \( C = F = F^* = \emptyset \).

1(d). Partition the set of \( N \) customers into two groups—\( G_1 \) containing all customers requiring service on two or more days and \( G_2 \) containing all customers requiring service on only one day.

1(e). Compute an expansion factor \( E = |G_1|/\mu_{\text{daily}} \), where \( \mu_{\text{daily}} \) is the mean number of stops required on each day and \( |G_1| \) is the number of customers in the template. Make an initial estimate for the maximum capacity of the template routes by setting \( Q_{\text{template}} = Q \times E = Q_0 \) and estimate the maximum travel time for the template routes by setting \( T_{\text{template}} = T/\sqrt{E} = T_0 \).

1(f). For all customers in \( G_1 \), set the demand amount and service time to be the mean values of these quantities taken across all days that the customer requires service.

Step 2. Initial set of template routes created.

2(a). Generate an initial set of template routes \( C \) for the customers in \( G_1 \) using the modified Clarke and Wright algorithm with parameter \( \lambda[l] \), vehicle capacity \( Q_{\text{template}} = Q_0 \), and maximum travel time \( T_{\text{template}} = T_0 \).

2(b). For each day \( d \), create routes by removing customers from \( C \) who do not require service on day \( d \) and then inserting customers from \( G_2 \) who require service only on day \( d \).

2(c). If the routes for all \( D \) days are feasible, set \( F = C, Q_{\text{old}} = Q_{\text{template}} \), and \( T_{\text{old}} = T_{\text{template}} \). Go to Step 3.

2(d). If at least one route on the \( D \) days is not feasible, then calculate the mean capacity violation \( (V_Q) \) and mean travel time violation \( (V_T) \) across all routes, and tighten the template constraints by setting

\[
Q_{\text{template}} = Q_{\text{template}} - V_Q/2 \quad \text{and} \quad T_{\text{template}} = T_{\text{template}} - V_T/2.
\]

Return to Step 2(a) and try to generate a set of feasible template routes.

Step 3. Diversification Phase. Modify the current feasible template routes \( C = F \). If \( f(C) < f(F^*) \), set \( F^* = C \).

3(a). Set \( \text{Record} \) equal to the total travel time of all routes in the current template \( C \). Set \( \text{Deviation} = \alpha \times \text{Record} \).

3(b). For \( i = 1 \) to \( I \)

Apply one-point move, two-point move, and two-opt move with RTR travel to the current template routes \( C \). Accept any improving move and only those
deteriorating moves where the total travel time of all template routes is less than \( \text{Record + Deviation} \) and where all routes satisfy the template constraints.

**Step 4.** Improvement Phase. Improve the current solution \( C \). Set \( k = 0 \).

4(a). Apply the one-point move, two-point move, and two-opt move, accepting only improving moves until no further improvements can be found.

4(b). Construct routes for each of the \( D \) days by applying the customer removal and insertion procedures.

4(c). If the routes for all \( D \) days are feasible, set \( F = C \). Compute the minimum slack amount across all daily routes in terms of capacity \( (S_Q) \) and travel time \( (S_T) \). Relax the template constraints by setting \( Q_{\text{template}} = Q_{\text{template}} + S_Q/2 \) and \( T_{\text{template}} = T_{\text{template}} + S_T/2 \). Go to Step 5.

4(d). If at least one route on the \( D \) days is not feasible, then compute the mean violations \( (V_Q \text{ and } V_T) \) as in Step 2(f) and tighten the template constraints by setting \( Q_{\text{template}} = Q_{\text{template}} - V_Q/2 \) and \( T_{\text{template}} = T_{\text{template}} - V_T/2 \). Set \( k = k + 1 \). If \( k < 5 \), continue to find feasible improvements by returning to Step 4(a). Otherwise, return to the last known feasible template, setting \( C = F \), and go to Step 5.

**Step 5.** If \( f(C) < f(F^*) \), set \( F^* = C \). If the objective function value of the current template routes \( C \) is less than \( \text{Record} \), set \( j = 0 \). Set \( j = j + 1 \). If \( j < J \), return to Step 3 and continue modifying the template. Otherwise, we have been unable to improve the current solution for \( J \) iterations, so stop modifying the current template and go to Step 6.

**Step 6.** Set \( I = I + 1 \) and return to Step 2 to generate a new initial solution if \( I \leq 3 \). Otherwise, use the best set of template routes \( (F^*) \) found during the search to generate the routes for each of the \( D \) days and return.

Before discussing the computational performance of our algorithm, we highlight a few of its more important features. The initial template routes that lead to feasible daily routes are generated by the modified Clarke and Wright algorithm using the \( \lambda \) parameter proposed by Yellow (1970). We make initial estimates of the template travel time and capacity limits in Step 1(e) by comparing the number of template customers with the average number of stops per day and then computing an expansion factor \( E \). After a template is found that leads to feasible routes for all \( D \) days, we run an improvement procedure and periodically alter the constraints on the template routes when we encounter daily routes that either violate the actual constraints or have significant slack. If a daily route violates a capacity limit or a travel time limit, then the offending limit (either \( Q \) or \( T \)) is decreased by one-half the mean violation amount (see Steps 2(d) and 4(d)). When the template is regenerated using this constraint, the idea is that because the template satisfies a tighter constraint, the daily routes will then be more likely to obey the actual constraints. In contrast, if we have slack in all the daily routes, then we increase each limit by one-half the mean amount of slack (see Step 4(c)). This allows for more flexibility in the improvement operations and will hopefully lead to better solutions as the template is modified. At the end of ConRTR, we return to the template that led to the best solution across the \( D \) days and return these routes. Because the routes for each day are constructed from the template routes, the precedence principle holds on these final routes—any two customers requiring service on the same day more than once during the \( D \) days will be served in the same order and will be serviced by the same driver.

### 4. Computational Experiments

In this section, we conduct several computational experiments that are designed to test the performance of ConRTR. First, we solve a set of small problems with 10 or 12 customer locations and compare the ConRTR solutions to the optimal solutions found by solving the MIP given in §2. Second, we solve 40 randomly generated problems with 1,000 customer locations and examine the performance of ConRTR as we change the customers’ service probabilities. Third, we construct a set of test problems from existing problems in the literature and solve them using ConRTR. Finally, we solve a problem with 3,715 customer locations that is based on five weeks of actual customer data provided by a firm in the small package shipping industry. We coded ConRTR in C++, and all experiments were conducted on a machine with a 1.4 GHz Intel processor and 512 MB of RAM.

#### 4.1. Small Problems

We generated five problems with 10 customer locations and five problems with 12 customer locations using the parameters below. The problem instances
and the optimal solutions are given in an online appendix and can be downloaded from http://www.rhsmith.umd.edu/faculty/bgolden/vrp_data.htm.

- Customer locations are generated uniformly at random within the square with vertices (0,0), (10,0), (10,10), (0,10).
- Travel times are defined to be the Euclidean distances.
- The depot is located at (0,0).
- There are three days of known service requirements.
- Customers require service on each day with probability 0.7.
- For customers requiring service on a given day, demand is uniformly distributed on [1, 3], and all service times are set to one unit.
- The maximum travel time is $T = 30$, and vehicle capacity is $Q = 15$ for all problems.
- The maximum arrival time differential is $L = 5$.

We compared the solutions produced by ConRTR on these 10 problems to the optimal solutions found by solving our MIP. ConRTR does not explicitly account for the maximum arrival time differential $L$. Instead, it relies on the precedence principle to address this aspect of consistent service. To account for this, we assessed the performance of ConRTR in two ways. First, we solved each problem using ConRTR and calculated the largest arrival time differential ($L_{\text{max}}$) across all customers and then solved the MIP by setting $L = L_{\text{max}}$. Next, we ran ConRTR again but discarded all solutions that had $L > 5$ and compared the remaining solutions to the optimal solutions produced by solving the MIP with $L = 5$.

In Table 2, we give the results produced by ConRTR and the MIP to these 10 problems. Columns 2 and 3 give the ConRTR and optimal solutions for the case of a fixed maximum arrival time differential of $L = 5$. The fifth and sixth columns give the best solution found by ConRTR when we ignore the $L = 5$ constraint along with the observed maximum arrival time differential $L_{\text{max}}$. The seventh column lists the optimal solutions to the MIP with $L = L_{\text{max}}$. Note that in several cases $L_{\text{max}} < 5$, so that the ConRTR solution satisfies the $L < 5$ constraint. The computing times for the MIP were very long, requiring up to several days using a state-of-the-art MIP solver (CPLEX 11.0), but ConRTR took less than one second to solve each problem. Nevertheless, the gap between the heuristic and optimal solutions is generally quite small, and ConRTR finds the optimal solution in 14 of the 20 instances.

One interesting feature of the optimal solutions is that for all problems except one, the optimal solution adheres to the precedence principle and can be generated from a set of template routes. In other words, there is only one case where customers who require more than one visit over the three days are visited in a different order.

### 4.2. Large Simulated Problems

Next, we generated a set of random problems to assess ConRTR under a number of different scenarios. In particular, we were interested in the effect of the frequency of service requirements on its performance. For example, if all customers required service every day, then the template itself could be traversed by the vehicles, leading to a situation in which the customers would be serviced at identical times each day. However, if customers required service with lower probabilities on each day, then the performance of ConRTR would suffer because of the larger number of modifications to the template as customers are inserted and removed when constructing the routes for each day.

In generating these problems, we set the travel time (in minutes) between two customers equal to the Euclidean distance. The service time was 1 minute and the maximum daily travel time for a vehicle was 500 minutes (this was based on discussions with the firm). The demand at each customer was uniformly distributed on [0, 10], and the vehicle capacity was 500 units. To produce problems that would lead to

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>ConRTR solution $L = 5$</th>
<th>Optimal solution $L = 5$</th>
<th>Gap (%)</th>
<th>ConRTR solution $L_{\text{max}}$</th>
<th>Optimal solution $L = L_{\text{max}}$</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>142.03</td>
<td>142.03</td>
<td>0</td>
<td>142.03</td>
<td>142.03</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>121.07</td>
<td>121.07</td>
<td>0</td>
<td>121.07</td>
<td>121.07</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>149.41</td>
<td>149.41</td>
<td>0</td>
<td>149.41</td>
<td>149.41</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>150.89</td>
<td>150.89</td>
<td>0</td>
<td>150.89</td>
<td>150.89</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>140.23</td>
<td>132.31</td>
<td>5.99</td>
<td>130.77</td>
<td>123.77</td>
<td>130.77</td>
</tr>
</tbody>
</table>
routes similar to those found by a package delivery company on a typical day over $D = 5$ days of known service requirements, we generated customer locations randomly, according to a uniform distribution, in a quarter circle with a radius of 80. With the distribution of customers, travel time, and capacity restrictions specified above, there were 100–150 customers on a route, which is typical for a package delivery company.

First, we developed 35 homogeneous test problems. Here, we have only one type of customer, and this customer is visited on each of the five days with a fixed probability $p$. We varied $p$ from 0.6 to 0.9 in steps of 0.05 and generated five homogeneous test problems with 700 customers for each value of $p$. Second, we developed five heterogeneous test problems. Here, we envision two types of customers—commercial and residential. The commercial customers have a high probability of being visited each day, and residential customers have a much lower probability of being visited each day. Consistent service by the same driver at the same time is not as important to residential customers (many are not present to receive a delivery) as it is to commercial customers. (We also point out that there is a very low probability that residential customers require service on more than one day.) We generated five heterogeneous test problems with 1,000 customer locations for each problem. Based on discussions with the firm, each heterogeneous test problem has 70% commercial customers (there is a 0.9 chance that a commercial customer is visited each day) and 30% residential customers (there is a 0.1 chance that a residential customer is visited each day). For the five heterogeneous problems, there were an average of 659 commercial and residential customers who required service on a given day.

For both sets of test problems, we ran ConRTR to generate consistent routes, and we calculated the mean maximum arrival time differential, the overall maximum arrival time differential, and the total travel time of the routes. Also, to get a sense for the increase in travel time resulting from the consistency constraints, we ran a generic RTR travel algorithm on each problem. This algorithm generally produces solutions that are within 1%–2% of the best known solutions on standard benchmark problems (Li 2005). By comparing the total travel time of the ConRTR routes to the total travel time of the RTR routes, we get a sense of the increased cost added by the consistency constraints. In particular, we compare the total number of vehicles and the total travel time required in the different solutions. We point out that ConRTR and RTR each take roughly 1–2 minutes to generate the routes for all five days.

In Table 3, we show the results produced by ConRTR and RTR on the 35 homogeneous test problems.

<table>
<thead>
<tr>
<th>$p$</th>
<th>Problem set</th>
<th>ConRTR results</th>
<th>RTR results</th>
<th>Average maximum time (min.)</th>
<th>Number of routes</th>
<th>Average number of routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td>1</td>
<td>11</td>
<td>6,717</td>
<td>5</td>
<td>6,312</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>9</td>
<td>6,691</td>
<td>5</td>
<td>6,472</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8</td>
<td>6,777</td>
<td>5</td>
<td>6,458</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>6,764</td>
<td>5</td>
<td>6,364</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>7,026</td>
<td>5</td>
<td>6,523</td>
<td>4.8</td>
</tr>
<tr>
<td>0.65</td>
<td>1</td>
<td>10</td>
<td>7,463</td>
<td>5</td>
<td>6,576</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>9</td>
<td>7,397</td>
<td>5</td>
<td>6,624</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8</td>
<td>6,982</td>
<td>5</td>
<td>6,655</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>7,349</td>
<td>5</td>
<td>6,800</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7</td>
<td>7,498</td>
<td>5</td>
<td>6,683</td>
<td>5.0</td>
</tr>
<tr>
<td>0.70</td>
<td>1</td>
<td>7</td>
<td>7,441</td>
<td>6</td>
<td>7,383</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7</td>
<td>7,900</td>
<td>6</td>
<td>7,049</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9</td>
<td>7,529</td>
<td>6</td>
<td>7,188</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>7,714</td>
<td>6</td>
<td>7,110</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7</td>
<td>7,990</td>
<td>6</td>
<td>7,171</td>
<td>5.2</td>
</tr>
<tr>
<td>0.75</td>
<td>1</td>
<td>7</td>
<td>7,638</td>
<td>6</td>
<td>7,160</td>
<td>5.6</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8</td>
<td>7,995</td>
<td>6</td>
<td>7,271</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>7,767</td>
<td>6</td>
<td>7,519</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7</td>
<td>7,558</td>
<td>6</td>
<td>7,400</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>7</td>
<td>7,967</td>
<td>6</td>
<td>7,430</td>
<td>6.0</td>
</tr>
<tr>
<td>0.80</td>
<td>1</td>
<td>5</td>
<td>8,226</td>
<td>6</td>
<td>7,701</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>8,391</td>
<td>6</td>
<td>7,751</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td>7,924</td>
<td>6</td>
<td>7,691</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>8,225</td>
<td>6</td>
<td>7,736</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>8,347</td>
<td>6</td>
<td>7,613</td>
<td>6.0</td>
</tr>
<tr>
<td>0.85</td>
<td>1</td>
<td>6</td>
<td>8,501</td>
<td>7</td>
<td>8,046</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>8,695</td>
<td>7</td>
<td>8,093</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>8,642</td>
<td>7</td>
<td>8,125</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>8,557</td>
<td>7</td>
<td>8,324</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>8,280</td>
<td>7</td>
<td>8,114</td>
<td>6.4</td>
</tr>
<tr>
<td>0.90</td>
<td>1</td>
<td>6</td>
<td>9,252</td>
<td>7</td>
<td>8,316</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>8,652</td>
<td>7</td>
<td>8,269</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>8,601</td>
<td>7</td>
<td>8,414</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>8,776</td>
<td>7</td>
<td>8,265</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>8,526</td>
<td>7</td>
<td>8,342</td>
<td>7.0</td>
</tr>
</tbody>
</table>
The results suggest that our precedence principle does a very good job preserving consistent arrival times for those customers requiring service on multiple days. For all 35 problems, taking into account the consistency requirements (denoted by ConRTR results in Table 3), the average maximum arrival time differential for all 700 customers was between 4 minutes ($p = 0.9$, problem sets 2-5) and 11 minutes ($p = 0.6$, problem set 1), with an overall average of 7 minutes. In the worst case, for one customer, the maximum arrival time differential was between 9 minutes ($p = 0.9$, problem set 3) and 50 minutes ($p = 0.65$, problem set 1), with an overall average of 22 minutes. In other words, using the routes generated by ConRTR, when a customer requires service on multiple days, the same driver visited a customer within 7 minutes (on average) of the same time each day. In the worst case (for one customer), this time increased to 22 minutes (on average).

Over all 35 problems, the total travel time of the ConRTR-generated routes was slightly longer than the total travel time of the routes generated by RTR. On average, the total travel time of the consistent routes was 6.6% longer (the increased times were between 0.8% and 13.5%). Additionally, on several occasions, RTR was able to produce solutions requiring one less vehicle. However, unless all of a particular driver’s customers do not require service on a particular day, the consistent driver constraint implies that we must always have the same number of vehicles operating each day. As expected, we observe that, for the routes produced by ConRTR, the average arrival time differential decreases as $p$ increases because the template routes account for more and more customers and are therefore able to do a better job of approximating each daily route.

In Table 4, we show the results produced by ConRTR and RTR on the five heterogeneous test problems. For all five problems, taking into account the consistency requirements, the average maximum arrival time differential for all commercial customers who require service on more than one day (recall that consistency is more important to the commercial customers than to the residential customers) was between 5 and 8 minutes, with an overall average of 7 minutes. In the worst case (for one customer), the maximum arrival time differential was between 16 and 43 minutes, with an overall average of 26 minutes. In other words, using the routes generated by ConRTR, when a customer requires service on multiple days, the same driver visited a customer within 7 minutes (on average) of the same time each day. In the worst case, this time increased to 26 minutes (on average).

Over all five problems, the total travel time of the ConRTR routes was slightly longer than the total travel time of the routes generated by RTR. On average, the total travel time of the consistent routes was 9.3% longer than the total travel time of the inconsistent routes (the increased times were between 6.5% and 11.3%). Finally, we note that for these five instances, ConRTR and RTR always generated solutions that contained seven routes.

### 4.3. Modified Benchmark Problems

There are a number of well-known benchmark problems for the classical VRP (see Christofides and Eilon 1969, Golden et al. 1998, Li et al. 2005). Given an $n$-node VRP benchmark problem, some number of days $D$, and a service probability $p$, we developed a simple procedure to randomly generate a ConVRP benchmark from this problem. We took the VRP benchmark problems of Christofides and Eilon (1969) and constructed a single five-day ConVRP benchmark from problems 1–12 using a daily service probability of $p = 0.7$ (these problems are available at http://www.rhsmith.umd.edu/faculty/bgolden/vrp_data.htm).

In Figure 2, we show the template and five days worth of routes for problem 3 with 100 nodes. In Table 5, we present the solutions found by our algorithm as well as the best total route length found.
Figure 2 ConRTR Solution to New Benchmark Problem 3

(a) Template
95 nodes 795.39 7 routes

(b) Day 1
70 nodes 740.20 7 routes

(c) Day 2
59 nodes 704.58 7 routes

(d) Day 3
72 nodes 721.19 7 routes

(e) Day 4
63 nodes 722.27 7 routes

(f) Day 5
73 nodes 739.98 7 routes
When consistency is not taken into account. We also include the mean maximum arrival time differential, as well as the overall largest maximum arrival time differential for the routes created by ConRTR.

The solutions presented in Table 5 are different from those in Table 3 in several ways. First, it is difficult to assess the consistency of the arrival times, as we have a much larger disparity in the total travel times and in the arrival time differentials from one problem to the next. For example, we have average maximum arrival time differentials ranging from 3.3 in problem 5 to more than 22 in problems 8 and 9. However, if we view the arrival time differential in proportion to the average route duration, then the average maximum arrival time differential is less than 10% in all 12 problems.

A second way that these solutions differ from those in Table 3 is that the solutions generated by RTR without regard for consistency generally require fewer vehicles and require roughly 15% less total travel time. Because the routes created by RTR are generally less costly than the ConRTR routes, we ran an experiment to transform the RTR solution into a solution that tries to satisfy the consistency constraints. To do this, we first had to assign a single driver to each route on each of the $D$ days so that as many customers as possible received consistent service throughout the $D$ days. Given this assignment of drivers to routes, we then had to traverse each route in one of the two possible ways (i.e., clockwise or counterclockwise) to minimize the mean maximum arrival time differential for the customers that always received service from the same driver.

The RTR-generated solutions sometimes require a different number of routes (drivers) on each day, so it is not clear how to analyze the consistency of these solutions. One option is to assign multiple drivers to individual vehicles on those days that require fewer routes. Another option is to have some drivers only work certain days. We chose to analyze the second option, reasoning that a firm would be unlikely to pay for more than a single driver per vehicle. Assuming that we have $v$ total drivers on each day (those who do not work on a particular day are assigned a trivial depot-to-depot route), we then can find the optimal assignment of drivers to routes in the $D$-day RTR solution by solving a set partitioning problem with an additional constraint. The details of this set partitioning problem are given in an online appendix.

After assigning drivers to routes, we next try to provide consistent service in terms of the arrival time differential. Each of the $v$ vehicles traverses $D$ routes, so there are $2^D$ orientations of these routes. By examining all orientations, we are able to find the optimal orientation that minimizes the mean maximum arrival time differential over those customers who receive service from only one driver. We present the results of these computations in Table 6.

By assigning drivers to routes in this way, we are able to provide consistent service to between 39% and 68% of the customers in Table 6, with an average of 50% (we are only able to provide an upper bound for problems 5, 9, and 10 because of the size of the set partitioning problem). In terms of the arrival time differentials, the mean maximum arrival time differential
is more than 50% larger, and the overall maximum arrival time differential is larger by a factor of more than 4.5. Thus, given efficient, inconsistent routes, this experiment suggests that it may not be possible to transform these routes into consistent ones.

### 4.4. Real-World Problem

Finally, we tested ConRTR on a real-world data set. This data set contained service requirements, demand amounts, and travel times for 3,715 customer locations over five weeks (five days per week). Although some of the customer locations had hard time windows, these were ignored because ConRTR does not account for this type of constraint.

We have two key interests in solving the real-world problem with ConRTR: (1) test the effectiveness of the precedence principle in handling actual service requirements and (2) examine the consistency of the arrival times at individual customers.

Important properties of the data set are summarized in Table 7. The second column gives the mean number of stops per day and the remaining columns describe the distribution of customer stops per week. Although most customers are visited on only one day \((k = 1)\), there is a significant number of customers requiring service on every day of the week \((k = 5)\). The right-most column gives the number of customers in the template (these customers require service on two or more days per week). A typical customer had a daily demand of three packages (some commercial customers required up to 90 packages), with a mean service time of three minutes. In general, the template contained roughly 85% as many customers as serviced on a typical day, implying that more customer insertions than deletions will be made to the template.

After reviewing the firm’s routes, we set the maximum total travel time to 9.5 hours and the total vehicle capacity to 350 packages. We applied ConRTR to each of the five weeks and generated routes for each day. As in previous experiments, we used RTR to generate routes that did not consider consistent service. In Table 8, we report the results of this experiment. For the consistent routes, the average maximum arrival time differential was between 19 and 35 minutes, with an overall average of 26 minutes. In the worst case (for one customer), the maximum arrival time differential was between 64 and 176 minutes, with an overall average of 101 minutes. With the exception of the routes for week 4, ConRTR performed quite well, with customers being visited within 25 minutes (on average) of the same time each day. In the worst case, this time increased to 83 minutes (on average). The consistency of the routes for the fourth week suffered because of a larger number of insertions on one day (causing late arrival times at some template customers) and a large number of deletions on another day (causing early arrival times for these same template customers).

#### Table 6: Assigning Drivers to Routes in the Optimal Way

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of customers</th>
<th>Proportion of customers receiving consistent service</th>
<th>Arrival time differentials</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>0.48</td>
<td>14.60</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>0.41</td>
<td>8.56</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0.48</td>
<td>11.22</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>0.46</td>
<td>8.65</td>
</tr>
<tr>
<td>5</td>
<td>199</td>
<td>≤0.46</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>0.55</td>
<td>23.42</td>
</tr>
<tr>
<td>7</td>
<td>75</td>
<td>0.39</td>
<td>11.83</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>0.47</td>
<td>22.34</td>
</tr>
<tr>
<td>9</td>
<td>150</td>
<td>≤0.51</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>199</td>
<td>≤0.49</td>
<td>—</td>
</tr>
<tr>
<td>11</td>
<td>120</td>
<td>0.68</td>
<td>15.17</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>0.54</td>
<td>4.82</td>
</tr>
</tbody>
</table>

#### Table 7: Properties of the Five-Week Data Set

<table>
<thead>
<tr>
<th>Week</th>
<th>Mean number of stops per day</th>
<th>Number of customers with (k) stops</th>
<th>Number of template customers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(k = 1)</td>
<td>(k = 2)</td>
</tr>
<tr>
<td>1</td>
<td>597</td>
<td>383</td>
<td>213</td>
</tr>
<tr>
<td>2</td>
<td>591</td>
<td>801</td>
<td>215</td>
</tr>
<tr>
<td>3</td>
<td>566</td>
<td>755</td>
<td>216</td>
</tr>
<tr>
<td>4</td>
<td>573</td>
<td>807</td>
<td>219</td>
</tr>
<tr>
<td>5</td>
<td>572</td>
<td>818</td>
<td>201</td>
</tr>
</tbody>
</table>

#### Table 8: Results for ConRTR and RTR on the Five-Week Data Set

<table>
<thead>
<tr>
<th>Week</th>
<th>ConRTR results</th>
<th>RTR results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average maximum differential</td>
<td>Overall maximum differential</td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>64</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
<td>176</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>85</td>
</tr>
</tbody>
</table>
The routes generated by RTR were slightly more efficient in terms of total travel time. However, the difference was only about 1%; this was quite different from the simulated problems, where the difference was generally about 9%. A visual inspection of these routes indicates that there is significant clustering in the data, and our template-based approach seems to work well with clustered data, as the insertion of customers into the template does not substantially increase the travel time.

In the final computational experiment with the five-week data set, we tested ConRTR's performance in generating a template for a future planning horizon using historical data. We produced a set of template routes from the first four weeks (20 days) of service requirements and then used this template to generate routes for each day of the fifth week. We modified the criterion for including a customer in the template, now requiring that a customer needs service on four or more days during this 20-day period. This criterion would allow us to provide consistent service to the most frequently visited customers by including them in the template routes.

The key point of this experiment is that the set of template routes was generated without ever considering the service requirements for the fifth week. Thus, we are trying to generate consistent routes for those customers who require service in the fifth week by using only the historical service requirements from the first four weeks of data. This allows us to more accurately model the situation in which a delivery company wishes to provide consistent service over a longer horizon than the fixed \(D\)-day period that we have previously considered.

In Table 9, we provide the results of the final experiment. We compare the routes derived from the historical template to the routes produced from the template that used the actual data from the fifth week. The second column gives the total travel time if we generate consistent routes by creating a set of template routes from the service requirements for the fifth week. The third column gives the total travel time of the routes that were created by constructing a template from the first four weeks of data and then removing and inserting customers, as determined by the service requirements for the fifth week. The final column gives the total travel time if consistency is disregarded and routes are generated for each day.

The results of this experiment are encouraging in terms of the efficiency of the routes that we were able to produce. Using the template generated from the first four weeks of data, the routes take 1% longer (on average) than the routes produced from the week 5 template. In terms of consistent arrival times, the maximum differential for customers requiring service on more than one day in week 5 was about 55 minutes. However, the overall maximum arrival time differential is much larger, at just over three hours. This large time is due to the fact that there were several customers in the week 5 data set who required service on two or more days but who were not included in the template routes, as they required service on fewer than four total days during the first four weeks. We expect that such times would decrease as the number of weeks of historical data increased.

Our results for the data set indicate that the service requirements from one week to the next are quite similar. Thus, after choosing an appropriate criterion for including customers in the template, an effective template that leads to high-quality routes can be generated solely from historical data.

### Table 9 Generating Consistent Routes from Historical Data

<table>
<thead>
<tr>
<th>Day</th>
<th>Total travel time of solution derived from week 5 template</th>
<th>Total travel time of solution derived from weeks 1–4 template</th>
<th>Total travel time of RTR-generated solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,190</td>
<td>1,197</td>
<td>1,183</td>
</tr>
<tr>
<td>2</td>
<td>1,132</td>
<td>1,136</td>
<td>1,119</td>
</tr>
<tr>
<td>3</td>
<td>1,147</td>
<td>1,164</td>
<td>1,136</td>
</tr>
<tr>
<td>4</td>
<td>1,133</td>
<td>1,137</td>
<td>1,124</td>
</tr>
<tr>
<td>5</td>
<td>1,226</td>
<td>1,265</td>
<td>1,214</td>
</tr>
</tbody>
</table>

5. Conclusions

In today’s highly competitive, small package shipping industry, companies are looking for ways to improve customer service. By providing consistent service, small package shipping companies can improve their relationships with customers by establishing a personal connection in the form of the same driver delivering packages at nearly the same time each day.

In this paper, we developed a method for generating consistent delivery routes. The routes produced by ConRTR were very successful in achieving customer service objectives with a low total travel time. ConRTR
performed well on randomly generated problems by producing routes on which drivers visited a customer within seven minutes (on average) of the same time each day. These consistent routes were only slightly longer in total travel time (less than 10% on average) than the corresponding inconsistent routes.

Furthermore, ConRTR performed well on four of five weeks of actual service requirements, with customers being visited within 25 minutes (on average) of the same time each day. The total travel time of the consistent routes was only about 1% greater than the total travel time of the inconsistent routes generated without regard for the additional customer service constraints. Finally, we demonstrated that ConRTR generated high-quality, consistent routes by using purely historical data.

We presented our results to the firm’s managers in late 2006. They found our approach novel and interesting. Our results confirmed that providing service consistency makes good sense.

Electronic Companion
An electronic companion to this paper is available on the Manufacturing & Service Operations Management website (http://msom.pubs.informs.org/ecompanion.html).

Acknowledgments
The authors thank the anonymous reviewers and the Associate Editor, in particular, for comments and suggestions that improved the paper. C. Gröer’s current affiliation is Oak Ridge National Laboratory, Oak Ridge, TN 37830.

References

In the paper “The Consistent Vehicle Routing Problem,” published in Manufacturing & Service Operations Management, ePub ahead of print December 4, 2008, http://msom.journal.informs.org/cgi/content/abstract/msom.1080.0243v1, the authors have amended the original text published online to correct an oversight in conveying the real-world problem studied in this article.