This article was downloaded by: [216.73.216.187] On: 14 July 2026, At: 15:54
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

Transportation Science

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

The Generalized Consistent Vehicle Routing Problem

Attila A. Kovacs, Bruce L. Golden, Richard F. Hartl, Sophie N. Parragh

To cite this article:
Attila A. Kovacs, Bruce L. Golden, Richard F. Hartl, Sophie N. Parragh (2015) The Generalized Consistent Vehicle Routing Problem. Transportation Science 49(4):796-816. https://doi.org/10.1287/trsc.2014.0529

This work is licensed under a Creative Commons Attribution 4.0 United States License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as "Transportation Science. Copyright 2015 INFORMS. http://dx.doi.org/10.1287/trsc.2014.0529, used under a Creative Commons Attribution License: http://creativecommons.org/licenses/by/4.0/us/"

Copyright © 2014, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

TRANSPORTATION SCIENCE
Vol. 49, No. 4, November 2015, pp. 796–816
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.2014.0529
© 2015 INFORMS

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

The Generalized Consistent Vehicle Routing Problem

Attila A. Kovacs
Department of Business Administration, University of Vienna, A-1090 Vienna, Austria,
attila.kovacs@univie.ac.at

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

Richard F. Hartl, Sophie N. Parragh
Department of Business Administration, University of Vienna, A-1090 Vienna, Austria,
{richard.hartl@univie.ac.at, sophie.parragh@univie.ac.at}

The consistent vehicle routing problem (ConVRP) takes customer satisfaction into account by assigning one driver to a customer and by bounding the variation in the arrival times over a given planning horizon. These requirements may be too restrictive in some applications. In the generalized ConVRP (GenConVRP), each customer is visited by a limited number of drivers and the variation in the arrival times is penalized in the objective function. The vehicle departure times may be adjusted to obtain stable arrival times. Additionally, customers are associated with AM/PM time windows. In contrast to previous work on the ConVRP, we do not use the template concept to generate routing plans. Our approach is based on a flexible large neighborhood search that is applied to the entire solution. Several destroy and repair heuristics have been designed to remove customers from the routes and to reinsert them at better positions. Arrival time consistency is improved by a simple 2-opt operator that reverses parts of particular routes.

A computational study is performed on ConVRP benchmark instances and on new instances generated for the generalized problem. The proposed algorithm performs well on different variants of the ConVRP. It outperforms template-based approaches in terms of travel cost and time consistency. For the GenConVRP, we experiment with different input parameters and examine the trade-off between travel cost and customer satisfaction. Remarkable cost savings can be obtained by allowing more than one driver per customer.

Keywords: vehicle routing; periodic distribution problems; large neighborhood search; metaheuristics
History: Received: May 2013; revision received: January 2014; accepted: January 2014. Published online in *Articles in Advance* June 17, 2014.

1. Introduction

Many routing problems have been investigated in the literature but very few of them take customer satisfaction into account. Customer-oriented routing problems arise in competitive industries (e.g., small package delivery) where customer satisfaction and customer loyalty are key to success. Another application appears in the home health care industry. Persons in need are visited at home to provide them with food, support in their daily hygiene, and medical treatment. The customers are highly sensitive to changes in their daily routine and in the staff that provides the service. In both examples, customer satisfaction can be increased by providing consistent service.

In the consistent vehicle routing problem (ConVRP) (Groër, Golden, and Wasil 2009) customers are serviced over a given planning period, e.g., several days. Customer satisfaction is maintained by providing driver and time consistent service. Driver consistency is expressed by the number of different drivers that visit a customer. Only one driver per customer is allowed in the ConVRP. Time consistency is achieved by bounding the maximum difference between the earliest and the latest arrival time at each customer. Vehicle idling to reduce the arrival time difference is not allowed. The objective is to find a minimum cost routing plan that complies with the classical routing constraints and the consistency requirements.

Since some assumptions in the ConVRP may be too restrictive in practice, we introduce the generalized consistent vehicle routing problem (GenConVRP). It extends and relaxes the ConVRP as follows:

- Each customer may be served by a limited number of drivers, instead of only one. This assumption is more realistic, since specific drivers are not always available (e.g., due to vacations or illnesses). In many cases, it is impossible to assign only one driver per customer over a long period of time. Typically, at least two drivers are regularly visiting a certain customer. Coelho, Cordeau, and Laporte (2012) propose a modified driver consistency feature where the number of different drivers per customer is minimized. They show that greater flexibility in the driver-to-customer assignment leads to slightly cheaper routing plans.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

- The maximum arrival time difference, i.e., time consistency, is not enforced by constraints but is penalized in the objective function. The number of drivers in the ConVRP is not restrictive, so it is always possible to generate a solution with a maximum arrival time difference of zero (e.g., if each driver visits only one customer). We include the maximum arrival time difference in the objective function because it is difficult to predict bounds that are sufficiently tight to guarantee high service quality without leading to a disproportional increase in travel cost.
- Customers are associated with AM/PM time windows. This constraint reflects the requirement of companies, which hire part-time workers to handle package deliveries (Wong 2008).
- Vehicle departure time is flexible. This extension is necessary because of the AM/PM time windows in combination with constraints that prohibit waiting times. Additionally, time consistency can be improved by adjusting the vehicle departure times (Kovacs, Parragh, and Hartl 2014).

Motivated by Ropke and Pisinger (2006b), Schrimpf et al. (2000), and Shaw (1998), we propose a large neighborhood search algorithm (LNS) for the GenConVRP. LNS is a local search technique that performs powerful moves in the search space by iteratively removing large parts of a solution and assembling a new and, hopefully, better solution.

The contribution of this paper is fourfold. First, we introduce the GenConVRP, which is formally defined in §3. Second, we devise a LNS metaheuristic tailored to the GenConVRP (§4). It integrates a new greedy algorithm to minimize the maximum arrival time differences at the customers by adjusting the vehicle starting times. Third, the LNS performs robustly for various versions and settings of the original ConVRP and dominates all previous (template-based) approaches, thus providing a new state of the art for this problem class. For the GenConVRP, we show that the gap with respect to the optimal solutions (on small instances that can be solved by CPLEX) is small. Finally, we examine the trade-off between travel cost and customer satisfaction on the basis of a variety of new and existing benchmark instances; the impact of a changing emphasis on time and driver consistency on variable and fixed costs (i.e., routing cost and fleet size) of service providers is analyzed in §5.

2. Related Literature

The difference between the classical vehicle routing problem and routing problems that have to be dealt with by companies in the small package shipping industry is described in Wong (2008). This industry is highly competitive and customers demand consistent visit schedules (e.g., a regular customer becomes accustomed to a specific delivery time and might be unsatisfied if there are large deviations from this time). Therefore, operations must be executed efficiently in terms of cost and in terms of customer satisfaction.

Appropriate quality measurements for dial-a-ride services for handicapped and elderly people are identified in Paquette et al. (2012). The authors conducted an extensive customer survey. Driver consistency was identified as one of the most important criteria to improve service quality.

The importance of service consistency in home health care is examined in Woodward et al. (2004). To provide high-quality service, the personnel (e.g., nurses, home workers, and physicians) must have knowledge and skills with regard to clients, homes, and services required. The personnel should know about likes and dislikes, culture, language, and routines of the clients. The clients must feel comfortable with the service provider in terms of physical intimacy. Personnel consistency is one of the key characteristics of a high-quality service and it reduces the complexity of communication between the involved persons. Time consistency permits the clients to plan their days in advance.

The consistent vehicle routing problem is presented in Groër, Golden, and Wasil (2009). It is a periodic delivery problem that integrates requirements of small package delivery companies into the vehicle routing problem. The objective is to minimize the total travel time. Hard constraints guarantee driver and time consistent service over a given planning horizon. The solution approach is based on the template concept. Here, daily vehicle routes are derived from a set of template routes. Templates can be used to generate solutions quickly in which each customer is visited by the same driver and in which customers are visited in the same sequence over several days. Templates that lead to high-quality solutions are generated by a record-to-record travel algorithm.

The template concept is also applied in Kovacs, Parragh, and Hartl (2014) and Tarantilis, Stavropoulou, and Repoussis (2012) to solve the ConVRP. In Tarantilis, Stavropoulou, and Repoussis, a tabu search algorithm is used to iteratively generate templates but also to improve the daily routes that are derived from the template routes. Templates are generated by adaptive large neighborhood search in Kovacs, Parragh, and Hartl.

An alternative approach to model time consistency is presented in Feillet et al. (2014). In contrast to the ConVRP, where the maximum arrival time difference of each customer is bounded, the authors minimize the maximum number of time classes in which a customer is visited. Visits are grouped into the same time class if the difference between the earliest and the latest visit is not larger than a given bound. Driver consistency is

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

not considered in the proposed solution approach, but the number of drivers per customer is reduced in a post-processing step.

In the courier delivery problem, presented by Sungur et al. (2010), routing decisions are made in a stochastic environment with random customer requests and random service times. The problem is solved by a master and daily scheduler (MADS). The idea is similar to the template concept: A robust master routing plan is generated by using scenarios obtained from historical demand data. Uncertainty is revealed at the beginning of each day. Then, the daily routes are derived from the master plan. Driver consistency is considered in the objective function. Customers have to be visited within given soft time windows.

Spliet and Gabor (2012b) and Spliet and Desaulniers (2012a) propose a routing problem in which a unique time window has to be assigned to each customer before demand is known. Stochastic information is given in the form of random scenarios. The objective is to find vehicle routes for each scenario that minimize the expected routing cost subject to the time window assignment.

The problem of clustering deterministic and stochastic customers into districts where each district is served by a single vehicle is proposed in Lei, Laporte, and Guo (2012). The solution approach is a two-stage program. Districting decisions are made in the first stage. In the second stage, the routing cost is approximated for each district.

Francis, Smilowitz, and Tzur (2007) investigate the effect of rigid versus flexible operating conditions (e.g., one driver per customer versus several drivers per customer, fixed delivery amounts to each customer versus variable amounts) on the solutions of the periodic vehicle routing problem (PVRP). Solutions are evaluated in terms of routing cost, variations in the arrival times at customers, variations in the regions a driver has to operate in, and variations in the drivers that visit the same customer. Smilowitz, Nowak, and Jiang (2013) also consider driver consistency in the context of the PVRP: in addition to routing costs, the number of different drivers per customer and the number of times a driver visits a different region are minimized.

Coelho, Cordeau, and Laporte (2012) and Coelho and Laporte (2013a, b) examine different approaches to integrate consistency into the inventory routing problem. The authors incorporate features that reduce, e.g., variations in the delivery quantity, variations in the number of drivers that provide the service, or variations in the time interval between consecutive deliveries.

3. Problem Definition

The GenConVRP is defined on a complete directed graph $G = (N, A)$, where $N = \{0, 1, \ldots, n\}$ is the set of customers and the depot $\{0\}$. Set $N^{am} \subseteq N\backslash\{0\}$ contains customers who can only be visited in the morning and set $N^{pm} \subseteq N\backslash\{0\}$ contains customers who require a visit in the afternoon ($N^{am} \cap N^{pm} = \{\}$); $A = \{(i, j) \mid i, j \in N,\ i \neq j\}$ is the set of arcs. (Arcs from customers $\in N^{pm}$ to customers $\in N^{am}$ can be omitted from the graph.) Customers are visited by a homogeneous fleet of vehicles in the set K. The number of vehicles is not restrictive (i.e., $|K| = |N\backslash\{0\}|$). Each vehicle $k \in K$ has a capacity of Q. Drivers and vehicles share a one-to-one relationship. We use drivers and vehicles interchangeably, depending on the context. Vehicles start and end their routes at the depot. The departure time is arbitrary, but vehicles must return to the depot before time T. The planning horizon involves $|D|$ days where D is the set of days. On each day $d \in D$, each customer $i \in N\backslash\{0\}$ has demand q_{id} and service time s_{id}. We use auxiliary parameters w_{id} equal to 1 if customer i requires service on day d ($q_{id} > 0$) and equal to 0, otherwise. Each arc $(i, j) \in A$ is associated with travel time t_{ij}. The travel time matrix satisfies the triangle inequality. Driver consistency is enforced by assigning each customer $i \in N\backslash\{0\}$ to at most W ($W \geq 1$) different drivers over the planning period. Parameter α is the weight of the total travel time and $(1 - \alpha)$ gives the weight of the maximum arrival time difference ($l_{\max}$) in the objective function. The model uses the following binary variables:

$$x_{ijkd} = \begin{cases} 1, & \text{if arc } (i, j) \text{ is traversed by vehicle } k \\ & \quad \text{on day } d, \\ 0, & \text{otherwise;} \end{cases}$$

$$y_{ikd} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to vehicle } k \\ & \quad \text{on day } d, \\ 0, & \text{otherwise;} \end{cases}$$

$$z_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to vehicle } k, \\ 0, & \text{otherwise.} \end{cases}$$

Continuous variables a_{id} denote the arrival time at customer $i \in N\backslash\{0\}$ on day d.

$$\text{Minimize} \quad \left\{ \alpha \sum_{d \in D} \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijkd} + (1 - \alpha) l_{\max} \right\} \tag{1}$$

$$\text{subject to} \quad y_{0kd} = 1 \quad \forall\ k \in K, d \in D, \tag{2}$$

$$\sum_{k \in K} y_{ikd} = w_{id} \quad \forall\ i \in N\backslash\{0\},\ d \in D, \tag{3}$$

$$\sum_{i \in N\backslash\{0\}} q_{id} y_{ikd} \leq Q \quad \forall k \in K,\ d \in D, \tag{4}$$

$$\sum_{i \in N} x_{ijkd} = \sum_{i \in N} x_{jikd} = y_{jkd} \quad \forall\ j \in N,\ k \in K,\ d \in D, \tag{5}$$

$$z_{ik} \geq y_{ikd} \quad \forall\ i \in N\backslash\{0\},\ k \in K,\ d \in D, \tag{6}$$

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

$$\sum_{k\in K} z_{ik} \le W \quad \forall i \in N\backslash\{0\}, \tag{7}$$

$$a_{id} + x_{ijkd}(s_{id} + t_{ij}) - (1 - x_{ijkd})T \le a_{jd} \quad \forall i, j \in N\backslash\{0\},\ k \in K,\ d \in D, \tag{8}$$

$$a_{id} + x_{ijkd}(s_{id} + t_{ij}) + (1 - x_{ijkd})T \ge a_{jd} \quad \forall i, j \in N\backslash\{0\},\ k \in K,\ d \in D, \tag{9}$$

$$a_{id} + s_{id} + t_{i0} \le T \quad \forall i \in N\backslash\{0\},\ d \in D, \tag{10}$$

$$(a_{i\alpha} - a_{i\beta})w_{i\alpha}w_{i\beta} \le l_{\max} \quad \forall i \in N\backslash\{0\},\ \alpha, \beta \in D, \tag{11}$$

$$a_{id} \le \frac{T}{2} \quad \forall i \in N^{am},\ d \in D, \tag{12}$$

$$a_{id} \ge \frac{T}{2} \quad \forall i \in N^{pm},\ d \in D, \tag{13}$$

$$a_{id} \ge t_{0i} \quad \forall i \in N,\ d \in D, \tag{14}$$

$$x_{ijkd} \in \{0, 1\} \quad \forall i, j \in N,\ k \in K,\ d \in D, \tag{15}$$

$$y_{ikd} \in \{0, 1\} \quad \forall i \in N,\ k \in K,\ d \in D, \tag{16}$$

$$z_{ik} \in \{0, 1\} \quad \forall i \in N,\ k \in K. \tag{17}$$

The objective function (1) minimizes the weighted average of the total travel time and the maximum arrival time difference. Inequalities (2) ensure that each route starts from the depot. Constraints (3) guarantee that each customer is serviced on each day he requires service. Equalities (4) limit the vehicle capacity to Q. Equalities (5) are flow conservation constraints. Driver consistency is ensured in (6) and (7). Constraints (6) set the z variables and constraints (7) make sure that the number of drivers per customer does not exceed the allowed limit. Inequalities (8) and (9) set the arrival times at the customers. Vehicle idling to improve time consistency is not allowed. Inequalities (8) also prevent subtours. Inequalities (10) enforce that vehicles return to the depot in time. Constraints (11) set the maximum arrival time difference. Time window feasibility is ensured in (12) and (13). Constraints (14) set the earliest possible arrival time at each customer. Finally, constraints (15)–(17) define the domains of the decision variables.

4. Solution Framework

A major difficulty in solving the ConVRP and its variants is the interrelation between the days in the planning horizon. State-of-the-art metaheuristics (Groër, Golden, and Wasil 2009; Tarantilis, Stavropoulou, and Repoussis 2012; Kovacs, Parragh, and Hartl 2014) rely on the template concept to generate good solutions. A template is a set of routes from which the daily routing plans are derived. On each day, the template is resolved by deleting customers that appear in the template but do not require service and by inserting customers that are not represented in the template but require service. Solutions derived from a template comply with driver consistency since the driver-to-customer assignment is not changed during the resolution. Additionally, the customer sequence in the template routes is transferred to the daily routes, which supports time consistency.

Our solution approach for the GenConVRP does not make use of template routes. The template concept restricts the search space and, therefore, good solutions are found quickly. Yet, better solutions might be missed. We propose a large neighborhood search (Shaw 1998) that is more flexible than template-based approaches because it operates on the entire routing plan. The pseudocode of the LNS is presented in Algorithm 1.

Given an initial solution, LNS iteratively destroys the current incumbent solution by removing a large number of customers. It then generates a new solution by using a repair operator to reinsert the removed customers into the routes (line 4). Several destroy and repair operators with different algorithmic structures are available. In each iteration, one destroy operator and one repair operator are selected randomly (line 3).

The vehicle departure times are adjusted in order to eliminate waiting times and to reduce the maximum arrival time difference (line 5). Time consistency of feasible solutions is improved by iteratively reversing parts of the route that contains the customer with the maximum arrival time difference (line 7). New solutions are accepted as current incumbents according to a simulated annealing criterion (line 9).

In the following, we describe the modules of the LNS algorithm in detail.

Algorithm 1 (LNS)

Require: initial solution s ▷ §4.6
1: $s^{\text{best}} = s$
2: **repeat**
3: select one destroy operator d and one repair operator r randomly
4: $s' =$ generateNewSolution(s, d, r)
5: adjustDepartureTimes(s') ▷ §4.3
6: **if** isDriverFeasible(s') **then**
7: improveTimeConsistency(s') ▷ §4.4
8: **end if**
9: **if** accept(s',s) **then** ▷ §4.5
10: $s = s'$
11: **end if**
12: **if** $f(s') < f(s^{\text{best}})$ **then**
13: $s^{\text{best}} = s'$
14: **end if**
15: **until** maximum number of iterations is reached
16: **return** s^{best}

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

4.1. Destroy Operators

We use several destroy operators to remove customers from the solution. Each of them adopts a different approach to improve the current solution. Customers are selected with regard to all days in the planning horizon and are removed from several days. Following Pisinger and Ropke (2007), the total number of removals, u, is chosen from the interval $[\min\{0.1\sum_{d\in D}\sum_{i\in N\setminus\{0\}} w_{id}, 30\}, \min\{0.4\sum_{d\in D}\sum_{i\in N\setminus\{0\}} w_{id}, 60\}]$. In large instances, the number of customers to remove is chosen from the interval $[30, 60]$; Pisinger and Ropke (2007) argue that removing less than 30 or more than 60 customers rarely leads to improving solutions.

We use five different destroy operators: random removal, related removal, and cluster removal are based on Shaw (1998), Pisinger and Ropke (2007), and Ropke and Pisinger (2006a, b). They are general operators for a variety of routing problems (a detailed description is given in Appendix A). Two versions of the worst removal operator are designed especially for the GenConVRP. In the following, we describe the new destroy operators in more detail.

Worst removal—driver consistency. The first worst removal operator uses a simple approach to reduce the number of different drivers per customer. Each customer i is associated with value $\mathscr{D}r(i)$. This value is composed of the number of drivers per customer z_i and customer i's arrival time difference l_i. It is given by

$$\mathscr{D}r(i) = z_i + \frac{l_i}{l_{\max} + \varepsilon}, \tag{18}$$

where ε is a small number > 0.

Customers are sorted in decreasing order of $\mathscr{D}r(i)$ (i.e., in a lexicographic order of z_i first and l_i second). Customers are then removed, one after another, from the solution until at least u removals have been made and all assigned customers comply with the driver consistency requirement (constraints 7).

Worst removal—time consistency. A second worst removal operator improves time consistency by repeatedly removing customers with large arrival time differences. The operator starts by dividing the days on which a customer requires service into two clusters. For each customer i, the arrival times are sorted in list L_i and the largest difference between two consecutive arrival times is identified. We obtain clusters, L_i^1 and L_i^2, by splitting L_i at the position with the largest difference in the arrival times. This approach is illustrated in Figure 1.

In a second step, we sort all customers in decreasing order of their maximum arrival time difference, l_i. Customers are then removed consecutively from the days in one of the clusters. Cluster L_i^1 is selected with probability $1 - |L_i^1|/|L_i|$ and cluster L_i^2 is selected with the complementary probability, $1 - |L_i^2|/|L_i|$. So, the more elements in a cluster, the lower the probability that it is selected for removal.

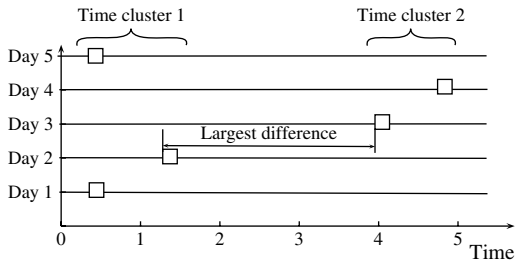


Figure 1 Time Clusters in Worst Removal Operator

Note. The squares represent the customer's arrival times on the respective days.

The operator stops when at least u removals have been made.

4.2. Repair Operators

Repair operators are used to insert unassigned customers into the routing plan. We use six operators that are either greedy based or regret based. They are motivated by Pisinger and Ropke (2007), and Ropke and Pisinger (2006a, b). The operators are applied to each day sequentially. Customers are inserted only with regard to the travel time and driver consistency. Considering time consistency on the basis of partial solutions is not easy. (This is why most algorithms use a template-based approach.) Our approach is to ignore time consistency until all requests have been assigned and to perform a post-processing step on the complete solution in order to reduce the maximum arrival time difference. Section 4.4 explains how the improvement of time consistency is accomplished.

The sequence in which days are repaired is random. This approach diversifies the search process because the customer-to-driver assignment on one day affects the feasibility of assignments on other days. We also randomize the process of inserting customers (Pisinger and Ropke 2007; Ropke and Pisinger 2006b). Therefore, we change a customer's insertion cost c to $c + y$; y is a randomly chosen number in the interval $[-\eta c, \eta c]$. Parameter η controls the scale of randomization. A repair operator is randomized with a probability of 50%.

In all repair operators, customers are scheduled at their earliest possible position with regard to the time windows.

Greedy operators. The applied greedy operators are based on Pisinger and Ropke (2007) and Ropke and Pisinger (2006b). The idea is to iteratively insert the customer that causes the least increase in the total travel time. Let c_{ik} be the change in the total travel time for inserting customer i at his cheapest position into route k; c_{ik} is set to infinity if an insertion violates the time windows, the maximum vehicle capacity Q,

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

or the allowed travel time T. A new vehicle is added to the solution if there is no feasible insertion position. In each iteration, we insert the customer with the lowest insertion cost (i^*) into his cheapest position. That is,

$$i^* := \underset{i \in N\setminus\{0\},\, k \in K}{\arg\min} \{c_{ik}\}. \tag{19}$$

We add a large penalty ($M = 1{,}000$) to c_{ik} if an insertion violates driver consistency. Furthermore, we reduce c_{ik} by M if i may not be assigned to additional drivers and k is already assigned to serve i on another day. In this way, we favor the insertion of customers that might be assigned to too many drivers in later iterations.

Two variants of the greedy operator are used: one as described above and one in which we artificially reduce the maximum vehicle capacity Q. The lower Q, the fewer customers are visited on a route, and the smaller is the arrival time difference. In the modified greedy operator, we select the artificial vehicle capacity randomly in the interval $[\max\{\max_{i\in N,\, d\in D}\{q_{id}\}, Q/2\}, Q]$. A lower interval boundary of $Q/2$ proved to work well during the implementation of the algorithm. However, the lower bound for the artificial capacity must be at least $\max_{i\in N,\, d\in D}\{q_{id}\}$ to ensure that all customers can be served.

Regret operators. Regret operators extend the greedy approach by taking into account the loss that might arise from postponing an insertion to later iterations (Potvin and Rousseau 1993). For each customer, we consider several insertion positions and insert the customer with the largest difference between his best insertion position and alternative positions. Let c_i^q denote the change in the total travel time for inserting customer i at his cheapest position into his q-cheapest route. In each iteration, customer i^* is inserted into his cheapest position according to

$$i^* := \underset{i \in N\setminus\{0\},\, k \in K}{\arg\max} \left\{ \sum_{h=2}^{\min(q,\, m)} (c_i^h - c_i^1) \right\}. \tag{20}$$

Parameter q defines the number of routes that are considered in the regret operator. Expensive insertions can be identified and avoided earlier with larger q values; m is the number of routes in the current solution.

We use four versions of the regret operator with $q = \{2, 3, 4, m\}$ as suggested in Pisinger and Ropke (2007) and Ropke and Pisinger (2006b). Driver consistency is taken into account by modifying the insertion costs as described above.

4.3. Adjustment of Vehicle Departure Times

In the GenConVRP, we may adapt the vehicle departure times from the depot. Flexible departure times are necessary because of the AM/PM time windows in combination with the constraints that prohibit waiting times (e.g., it is not possible to visit only PM customers if departure times are fixed at 0). Additionally, time consistency can be improved significantly by adjusting the vehicle departure times (Kovacs, Parragh, and Hartl 2014). The problem of adjusting the departure times of a given solution is modeled as follows.

$$\text{Minimize} \quad l_{\max} \tag{21}$$

$$\text{subject to} \quad a_{id} + s_{id} + t_{ij} = a_{jd} \quad \forall (i,j) \in A_{kd},\ i, j \in N\setminus\{0\},\ d \in D, \tag{22}$$

$$a_{id} + s_{id} + t_{i0} \le T \quad \forall i \in N\setminus\{0\},\ d \in D, \tag{23}$$

$$(a_{i\alpha} - a_{i\beta}) w_{i\alpha} w_{i\beta} \le l_{\max} \quad \forall i \in N\setminus\{0\},\ \alpha, \beta \in D, \tag{24}$$

$$a_{id} \le \frac{T}{2} \quad \forall i \in N^{am},\ d \in D, \tag{25}$$

$$a_{id} \ge \frac{T}{2} \quad \forall i \in N^{pm},\ d \in D, \tag{26}$$

$$a_{id} \ge t_{0i} \quad \forall i \in N,\ d \in D. \tag{27}$$

The objective is to minimize the maximum arrival time difference $l_{\max}$ (21). Constraints (22) set the arrival times at the customers. Set A_{kd} contains all arcs that are traversed by vehicle k on day d. Inequalities (23) guarantee that vehicles return to the depot in time. The maximum arrival time difference is set by inequalities (24). AM/PM time windows are enforced by inequalities (25) and (26). Constraints (27) defines the earliest possible arrival time at each customer.

The adjustment of the departure times can be performed by any linear programming (LP) solver. Yet, the LP becomes the bottleneck of the algorithm when it is integrated into the LNS. Instead of the LP, we use a fast greedy algorithm to reduce the maximum arrival time difference. The idea is to iteratively shift the departure time of the vehicle that visits the customer with the maximum arrival time difference. This approach is motivated by Kovacs, Parragh, and Hartl (2014). However, our algorithm is more sophisticated. In Kovacs, Parragh, and Hartl, the focus is on the customer that violates the maximum allowed arrival time difference the most; the arrival times of other customers are disregarded. Therefore, the algorithm might reduce the arrival time difference at one customer but increase $l_{\max}$ at another. In our algorithm, $l_{\max}$ never deteriorates from one iteration to the other; additionally, in our experiments, it has always provided the optimal solution. The pseudocode of our approach is given in Algorithm 2.

Algorithm 2 (*adjustDepartureTimes*)

Require: solution s

1: *eliminateWaitingTimes*(s)
▷ waiting times are not allowed

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

2: **repeat**
3: $i^{\max} = taskWithMaxATD(s)$
4: $BC = \{i^{\max}\}$ ▷ blocking customers
5: $\max PF = \max ATD(s)$
6: **for all** $j \in BC$ **do**
7: $\max PF = \min\{\max PF, getMaxPF(j, BC, s)\}$
8: **end for**
9: **if** $(\max PF > \varepsilon)$ **then** ▷ $\varepsilon = 10^{-4}$
10: $applyPF(\max PF, BC, s)$
11: **else** ▷ pushing forward is not possible ⇒ try to pull backwards
12: $BC = \{i^{\max}\}$
13: $\max PB = \max ATD(s)$
14: **for all** $j \in BC$ **do**
15: $\max PB = \min\{\max PB, getMaxPB(j, BC, s)\}$
16: **end for**
17: **if** $(\max PB > \varepsilon)$ **then**
18: $applyPB(\max PB, BC, s)$
19: **end if**
20: **end if**
21: **until** $\max PF > \varepsilon \vee \max PB > \varepsilon$

The repair operators insert customers at their earliest possible positions. We start by eliminating the waiting times between the last AM customer and the first PM customer on each route (line 1). If i is the last AM customer on route k on day d and customer j is the first PM customer on the same route, we delay the departure time of that route by $a_{jd} - (a_{id} + s_i + t_{ij})$.

In each iteration of the algorithm, the customer with the current maximum arrival time difference $i^{\max}$ is used to initialize set BC (lines 3 and 4); BC contains all customers that have to be moved in order to reduce $l_{\max}$. We call these customers blocking customers. For each customer in BC, we compute the amount of time he can be pushed forward without violating any constraint or increasing the current maximum arrival time difference (line 7).[1]

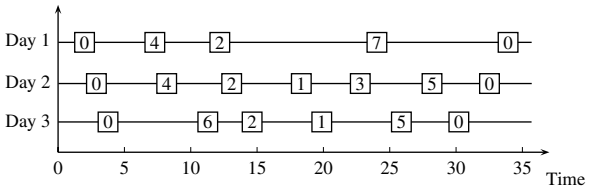
The computation is done in function $getMaxPF(j, BC, s)$. The function first identifies the day (or the days) on which customer j is visited earliest. The maximum push forward $pf(j, k)$ of j's route on the respective day (or days) is computed with respect to all other customers k on the same route. The upper bound for $pf(j, k)$ is the value that puts the arrival time difference of k on a level with the arrival time difference of j. Consider, for example, two routes 0–a–b–0 and 0–b–c–a–0 that are executed on two days, respectively; customer a has the largest difference in the arrival times. The algorithm reduces l_a by pushing forward the route on day one. This, however, increases l_b. The maximum push forward $pf(j, k)$ is limited by the value that sets $l_a = l_b$.

Customers that block the pushing forward of task j (i.e., $pf(j, k) < \varepsilon$) are added to set BC. We obtain the maximum push forward $\max PF$ by evaluating the push forward of all customers in set BC with respect to all customers that might block the shifting and then take the minimum. Details on how the $pf(j, k)$ values are computed are given in Appendix B.

By being greedy and shifting routes forward as much as possible, we can get trapped in a local optimum. Lines 12–18 of the algorithm serve as an escape mechanism. Effectively, we try to reduce the delay that we have applied before.

The algorithm stops when neither pushing routes forward nor pulling them backward can reduce the maximum arrival time difference.

Figure 2 illustrates how the algorithm works. The figure at the top shows the routing plan for each day in a three-day planning horizon; seven customers are visited. The three tables show three iterations of the algorithm. We report the arrival times at each customer on each day, the arrival time difference for each customer, $\max PF$, $\max PB$, and set BC. In iteration 1,



Iteration 1

Day	1	**2**	3	4	**5**	6	7
1		11.7		6.9			22
2	17.5	14.5	22.7	9.7	29.3		
3	18.7	15.7			25.3	12	
l_i	1.2	**4**	0	2.8	**4**	0	0

$i^{\max}$	5
BC	5, 2
maxPF/maxPB	0.2/-

Iteration 2

Day	1	**2**	3	4	5	6	7
1		11.9		7.1			22.2
2	17.5	14.5	22.7	9.7	29.3		
3	18.9	15.9			25.5	12.2	
l_i	1.4	**4**	0	2.6	3.8	0	0

$i^{\max}$	2
BC	2
max*PF*/max*PB*	0/0.1

Iteration 3

Day	1	**2**	3	4	**5**	6	7
1		11.9		7.1			22.2
2	17.5	14.5	22.7	9.7	29.3		
3	18.8	15.8			25.4	12.1	
l_i	1.3	**3.9**	0	2.6	**3.9**	0	0

$i^{\max}$	2
BC	2, 5
max*PF*/max*PB*	0/0

Figure 2 Three Iterations of Algorithm 2

[1] Waiting times are not allowed. So, shifting customer j's starting time on a given day is equivalent to shifting the departure time of the vehicle that visits j.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

customers 2 and 5 have an arrival time difference of 4; $i^{\max}$ is set to 5, arbitrarily. To reduce $l_{\max}$, customer 5's arrival time on day 3 has to be pushed forward. Customer 2 is a blocking customer because its latest arrival time is on day 3 and its arrival time difference is 4. The algorithm delays the routes on day 1 and day 3 by 0.2; a larger $\max PF$ would violate the shift length constraint on day 1. In iteration 2, $l_{\max}$ is still 4 at customer 2. So, $i^{\max}$ is set to 2. Reducing $l_{\max}$ by pushing routes forward is infeasible; we are in a local optimum. Therefore, the route on day 3 is pulled backward by 0.1. This reduces $l_{\max}$ to 3.9. In iteration 3, customers 2 and 5 have an arrival time difference of 3.9 but both $\max PF$ and $\max PB$ are 0; the algorithm stops.

4.4. Improvement of Time Consistency

The repair operators insert customers into the routing plan only with regard to the travel time. So, new solutions might be poor in terms of time consistency. If a new solution is feasible, we reduce the customers' arrival time differences by reversing parts of selected routes. This approach corresponds to repeatedly executing a 2-opt heuristic (Lin 1965) on the route that contains the customer with the current maximum arrival time difference. The pseudocode is presented in Algorithm 3.

Algorithm 3 (*improveTimeConsistency*)

Require: feasible solution s

1: **repeat**
2: $f' = f(s)$
3: $i^{\max} = taskWithMaxATD(s)$
4: $aat = averageArrivalTime(i^{\max}, s)$
5: $eat = earliestArrivalTime(i^{\max}, s)$
6: $lat = latestArrivalTime(i^{\max}, s)$
7: **if** $(aat - eat > lat - aat)$ **then**
8: $d = dayEarliestArrival(i^{\max}, s)$
9: **else**
10: $d = dayLatestArrival(i^{\max}, s)$
11: **end if**
12: $apply2opt(i^{\max}, d, s)$
13: **until** $f(s) < f'$ or we have a special case as shown in Figure 4

First, we identify the customer with the maximum arrival time difference $i^{\max}$ (line 3). Then, we use the earliest, average, and latest arrival times of $i^{\max}$ to select a day to reverse parts of the route that contains $i^{\max}$. By using the selection criterion in line 7, we select the day that has the highest chance to reduce the variance in the arrival times. This is the day on which customer $i^{\max}$ is visited either latest or earliest. The 2-opt heuristic is applied to the route that contains $i^{\max}$ on the selected day (line 12). A reversal may affect either only customers $\in N^{am}$ if $i^{\max} \in N^{am}$ or only customers $\in N^{pm}$ if $i^{\max} \in N^{pm}$.

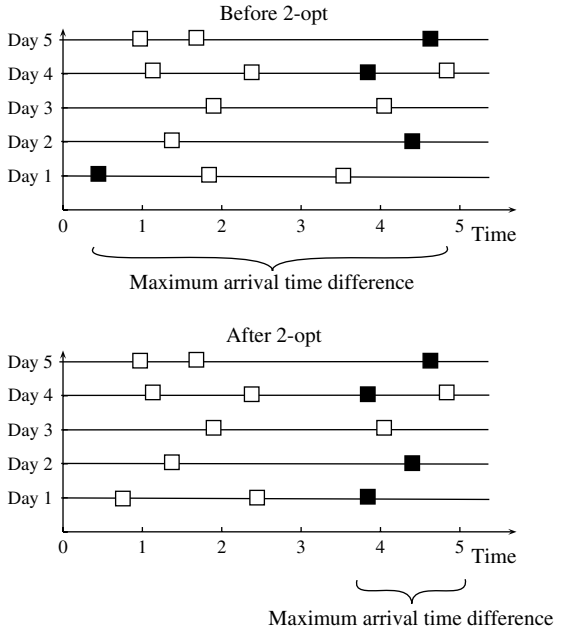


Figure 3 2-Opt Operator to Improve Time Consistency

Notes. The black squares represent the customer with the maximum arrival time difference. The white squares represent other customers.

A route that improves the solution is accepted as the basis for further reversals. Large reductions in the maximum arrival time difference can be expected by reversing long sequences. Therefore, it is important to start the search with the longest sequence and then reduce the length of the sequence until all 2-opt moves have been tried.

The desired result after one iteration is shown in Figure 3. The black squares represent the starting times of customer $i^{\max}$. The route selected for the 2-opt heuristic is the one that contains $i^{\max}$ on day 1. The procedure stops when no improvement is obtained by reversing parts of only one route. However, with this stopping criterion the algorithm fails to improve solutions like in Figure 4. In this example, the difference in the starting time on the selected day and at least one other day is close to zero. The solution would not improve by reversing only one route. In such cases, we ignore all days where $i^{\max}$ has the same arrival

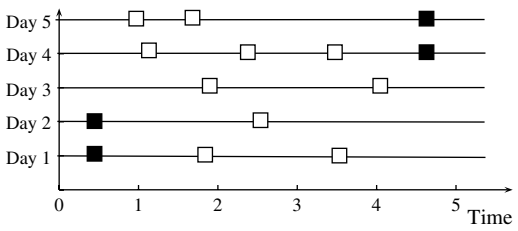


Figure 4 Special Case for 2-Opt Operator

Notes. The black squares represent the customer with the maximum arrival time difference. The white squares represent arbitrary customers.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

time as the day on which 2-opt is performed. With this modification, we can improve the solution step-by-step.

The 2-opt heuristic is costly in terms of computation time because the evaluation of one reversal involves the entire solution and requires the readjustment of vehicle departure times. In Appendix C, we describe an approach to speed up the algorithm by avoiding the execution of reversals that cannot lead to improved solutions.

4.5. Acceptance Criterion

The decision whether or not a new solution s' is accepted as the current incumbent s is based on a simulated annealing acceptance criterion (Kirkpatrick, Gelatt, and Vecchi 1983). The evaluation function $g(s)$ is composed of the objective function $f(s)$ and a penalty term for violations of the driver consistency. That is,

$$g(s) = f(s) + (e^{\max\{0,\, z^{\max} - W\}/p_{\text{penalty}}} - 1) i_{\text{LNS}}; \tag{28}$$

$z^{\max}$ is the maximum number of drivers per customer ($z^{\max} = \max_{i \in N \setminus \{0\}} \{z_i\}$; z_i denotes the number of different drivers per customer i). W is the number of allowed drivers per customer. Parameter p_{penalty} is set such that the penalty for violating the driver consistency by one driver in the last iteration, i_{LNSmax}, is equal to the objective value of the initial solution, s^{initial}. (The initial solution is feasible, i.e., $f(s^{\text{initial}}) = g(s^{\text{initial}})$.) This yields

$$p_{\text{penalty}} = \frac{1}{\ln(f(s^{\text{initial}})/i_{\text{LNSmax}}) + 1}. \tag{29}$$

Improving solutions (i.e., $g(x') < g(x)$) are always accepted as current incumbents. Using a simulated annealing rule, deteriorating solutions are accepted with probability $e^{-(g(s') - g(s))/\hat{t}}$. Parameter $\hat{t}$ is the current temperature in the annealing process. It is initialized by the term

$$\hat{t} = -\frac{w_{\hat{t}}}{\ln 0.5} f(s^{\text{initial}}). \tag{30}$$

Thus, a $w_{\hat{t}}$% worse solution is accepted with a probability of 50% (Ropke and Pisinger 2006b; Pisinger and Ropke 2007). The cooling rate is controlled by parameter c, i.e., $\hat{t}$ is replaced by $\hat{t}c$ in each LNS iteration.

The best found solution s^{best} is replaced if s' complies with all constraints and $f(s') < f(s^{\text{best}})$.

4.6. Initial Solution

Parameter α in the objective function has a strong impact on the structure of a good solution. A solution that visits each customer on a separate route is optimal if α is 0. If $\alpha = 1$, several customers are consolidated on each route. Because of this difference in the solution structure, better final solutions can be obtained by using a proper starting solution. This is especially true when α is very small and the maximum arrival time difference is close to 0 in the optimal solution.

We construct two initial solutions, one that is travel-time oriented and one that is time-consistency oriented. The LNS is started with the solution that has a lower objective value.

Travel-time oriented solution. We apply the greedy repair operator (with the original vehicle capacity) from §4.2 and the worst driver destroy operator from §4.1 to construct a feasible initial solution with reasonable total travel time. The algorithm iteratively applies the greedy operator[2] to insert customers into routes and the worst driver destroy operator to remove all customers that violate the driver consistency requirement. One of the removed customers is assigned to a new vehicle and the remaining customers are inserted by the greedy operator into their cheapest position. The process stops when the solution complies with the driver consistency constraint. Time consistency is improved as described in §4.4.

Time-consistency oriented solution. An initial solution with a low arrival time difference is generated by assigning each customer that requires at least two visits in the planning period to a separate route. Customers that require only one visit are then inserted by the greedy operator and time consistency is improved as described in §4.4.

4.7. Further Improvements

The LNS is designed for commercial applications where cost is the most important aspect. Therefore, in its pure version, the algorithm performs poorly if the focus is on time consistency rather than on travel time (i.e., α is very small). Experiments show that we can improve the time consistency by artificially restricting the search space by reducing the number of allowed drivers per customer. With this extension, we improve the time consistency without losing much travel time for a broad range of α settings.

The procedure is as follows. Regardless of the number of allowed drivers W per customer, we start the search (including the initial solution) by restricting the driver consistency to one driver per customer. The restriction applies only to the repair operators and the worst driver destroy operator—the evaluation function is not affected. We increase the number of allowed drivers by one if the best found solution has not improved for $i_{\text{LNSmax}}/5$ iterations (i_{LNSmax} is the maximum number of LNS iterations). In the last 5% of the search, we again allow W drivers per customer.

5. Computational Results

Experiments on different ConVRP variants are conducted by running a C++ implementation of the LNS

[2] Driver consistency is considered in the greedy operator. However, there is no guarantee that the allowed number of drivers per customer is not exceeded.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

Table 1 Adjusted Parameters

Simulated annealing		Repair operators	Related destroy operator			
w_t	c	η	λ	μ	ν	ξ
0.05	0.9999	0.1	12	4	4	7

on an Intel Xeon X5550 computer with 2.67 GHz. The number of LNS iterations is 50,000. Results are obtained by taking the average of 10 runs per instance.

LNS algorithms tend to be robust with respect to the chosen control parameters. Therefore, we kept the efforts to tune them low. Each parameter was set to either an initial guess or to a value reported in the literature (Ropke and Pisinger 2006b; Kovacs, Parragh, and Hartl 2014). Only the parameters listed in Table 1 were adjusted during the development of the algorithm. The same set of parameters is used for all experiments.

5.1. Data Sets

We apply the LNS to different variants of the ConVRP in order to examine its robustness. The characteristics of the respective data sets are described below.

Instances for ConVRP. A benchmark data set for the ConVRP is provided by Groër, Golden, and Wasil (2009). The data set consists of 12 instances with 50 to 199 customers and a planning horizon of five days. Customer requirements are fixed in advance for the entire planning period. The probability that a customer requires service on a given day was set to 70% when instances were generated. We refer to this probability as service frequency. The higher the service frequency the more customers are serviced at least twice, i.e., more customers demand consistent service. Furthermore, the daily routing plans become similar with higher service frequency. No bound on the maximum arrival time difference is given in the benchmark data set. We set the maximum arrival time difference according to the results obtained in Groër, Golden, and Wasil. We refer to the benchmark data set as data set A.

Instances for ConVRP with adaptable departure times. The original ConVRP instances are extended in Kovacs, Parragh, and Hartl (2014). The extension refers to the service frequency that was set to 50% and 90%, respectively; the service frequency in the original instances is 70%. Also, different bounds on the maximum arrival time difference are suggested. We use the instances with the tightest bounds since these are the most challenging. The vehicle departure time may be adjusted to improve time consistency. We refer to the data sets with 50%, 70%, and 90% service frequency as B_0.5, B_0.7, and B_0.9, respectively.

Instances for GenConVRP. ConVRP instances are turned into GenConVRP instances by adding AM/PM time windows. In each instance, we associate customers with even IDs with AM time windows and customers with odd IDs with PM time windows. The shift duration T is not bounded in all ConVRP instances. In instances without bounds on the shift length, we set T to 1,000. This constraint is very loose but we obtain reasonable time window ranges. The data sets with time windows are named C_0.5, C_0.7, and C_0.9, with respect to the service frequencies.

There are no reference solutions for the GenConVRP. Therefore, we also use 10 small test instances that can be solved by MIP solvers. The small instances contain 10 to 12 customers that are visited over a planning horizon of three days. They are derived from the small ConVRP instances presented in Groër, Golden, and Wasil (2009). We refer to the small GenConVRP data set as C_small.

Each instance is tested with different bounds on the number of allowed drivers (W) and with different parameters α in the objective function (i.e., different emphasis on time consistency).

5.2. Results for ConVRP Benchmark Instances (Data Set A)

The performance of the LNS on the ConVRP benchmark instances (data set A) is compared to three template-based algorithms: the record-to-record travel algorithm for the ConVRP (ConRTR) (Groër, Golden, and Wasil 2009), the template-based tabu search (TTS) (Tarantilis, Stavropoulou, and Repoussis 2012), and the template-based adaptive large neighborhood search (TALNS) (Kovacs, Parragh, and Hartl 2014). The results of the comparison are reported in Table 2. For each algorithm and each instance, we give the maximum arrival time difference ($l_{\max}$) and the total travel time plus the total service time[3] (TT). All algorithms are stochastic with the exception of the ConRTR. TT(avg) denotes the average results over 10 runs. TT(min) is the best result out of five runs for the TTS and out of 10 runs for the LNS. If the computation time is available we report it in seconds, CPU (s).

TT Imp (%) denotes the improvement in the total travel time that is obtained by the LNS compared to the respective algorithms. The improvement between LNS and TTS refers to TT(min), the improvement between the other algorithms to TT(avg). The average results over all instances are given in the last row of the table. The last column gives the improvement in the arrival time difference obtained by LNS compared to TALNS.

The objective in the ConVRP is the minimization of the total travel time while bounding the maximum arrival time difference. LNS is modified to check the maximum arrival time difference for feasibility before replacing the best found solution and parameter α is set to 0.3.

[3] We add the total service time to the total travel time to be consistent with the literature.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

Table 2 Comparison with Other Algorithms on Data Set A

	ConRTR		TTS		TALNS			LNS			*TT* Imp (%)			l_{max} Imp (%)
Instances	*TT* (avg)	l_{max}	*TT* (min)	l_{max}	CPU (s)	*TT* (avg)	l_{max}	CPU (s)	*TT* (avg)	*TT* (min)	l_{max}	CPU (s)	to ConRTR	to TTS	to TALNS	to TALNS
Christofides_1_5_0.7	2,282.14	24.38	2,210.56	21.99	80.00	2,194.93	23.72	5.45	2,132.47	2,124.21	20.18	15.13	6.56	3.91	2.85	14.94
Christofides_2_5_0.7	3,872.86	34.26	3,622.71	27.75	93.00	3,605.03	31.86	14.69	3,591.04	3,540.80	25.41	18.82	7.28	2.26	0.39	20.26
Christofides_3_5_0.7	3,628.22	22.87	3,451.10	21.92	369.00	3,338.03	22.21	25.58	3,310.20	3,280.47	21.22	40.22	8.77	4.94	0.83	4.44
Christofides_4_5_0.7	4,952.91	27.53	4,572.00	25.15	388.00	4,598.78	24.19	84.31	4,557.72	4,473.31	19.85	62.71	7.98	2.16	0.89	17.95
Christofides_5_5_0.7	6,416.77	26.93	5,732.62	19.99	550.00	5,685.55	22.69	122.24	5,675.16	5,632.22	17.69	87.26	11.56	1.75	0.18	22.04
Christofides_6_5_0.7	4,084.24	63.47	4,096.87	55.38	70.00	4,051.50	63.26	6.63	4,073.92	4,070.72	40.39	14.58	0.25	0.64	−0.55	36.15
Christofides_7_5_0.7	7,126.07	83.96	6,752.36	63.28	161.00	6,805.99	76.62	18.33	6,713.15	6,673.61	43.72	19.68	5.79	1.17	1.36	42.93
Christofides_8_5_0.7	7,456.19	73.04	7,279.39	62.01	539.00	7,192.45	65.97	32.24	7,197.32	7,126.29	50.27	31.27	3.47	2.10	−0.07	23.79
Christofides_9_5_0.7	11,033.54	106.43	10,585.10	84.76	947.00	10,450.04	88.85	97.39	10,444.95	10,390.70	59.07	50.24	5.33	1.84	0.05	33.52
Christofides_10_5_0.7	13,916.80	60.17	13,120.40	57.17	1,052.00	13,238.57	57.95	146.32	13,042.63	12,955.10	55.19	78.73	6.28	1.26	1.48	4.77
Christofides_11_5_0.7	4,753.89	16.10	4,721.09	15.68	480.00	4,506.59	15.33	35.96	4,588.43	4,471.22	13.91	83.63	3.48	5.29	−1.82	9.28
Christofides_12_5_0.7	3,861.35	17.58	3,607.88	16.91	172.00	3,530.16	16.50	25.60	3,585.73	3,521.88	13.63	27.41	7.14	2.38	−1.57	17.40
Average	6,115.42	46.39	5,812.67	39.33	408.42	5,766.47	42.43	51.23	5,742.73	5,688.38	31.71	44.14	6.16	2.48	0.34	20.62

The comparison of LNS to the template-based approaches shows the advantage of a nontemplate-based approach. LNS performs better than ConRTR and TTS on all instances with an average improvement of 6.16% and 2.48%, respectively. The difference in the total travel times is minor between LNS and TALNS. However, the arrival time difference is on average 20.26% lower with LNS than with TALNS. (The total travel time can be improved at the cost of a higher arrival time difference by increasing parameter α in the LNS.) For a sample of 10 runs, the average variation coefficient (standard deviation/mean) of the LNS is 0.0077; the reported variation coefficient for the TALNS is 0.0097 (Kovacs, Parragh, and Hartl 2014). This result supports previous findings that LNS, if properly designed, is a very robust metaheuristic. Additionally, with an average computation time of 44 seconds, LNS is faster than the other algorithms.

5.3. Results for ConVRP with Adaptable Starting Times (Data Sets B_0.5, B_0.7, and B_0.9)

Tables 3–5 compare the LNS and the TALNS when the vehicle departure time is adaptable. The departure times are adjusted by an exact approach in the TALNS and by Algorithm 2 (described in §4.3) in the LNS. LNS accepts only solutions that comply with the time consistency requirement as best found solutions; parameter α is set to 0.2. The tables present the average total travel time plus the total service time (TT_{avg}), the maximum arrival time difference (l_{max}), and the computation time (CPU (s)) for LNS and TALNS, respectively. The last two columns show the respective improvements in TT_{avg} and l_{max} of LNS over TALNS.

LNS performs well on this ConVRP variant despite tight bounds on l_{max}. An interesting pattern can be observed when comparing results with different service frequencies. LNS outperforms TALNS on data set B_0.5 with an average improvement of 1.42% in the total travel time and an average reduction of 16.41% in the maximum arrival time difference. The advantage of LNS diminishes to 0.67% with respect to *TT* and to 6.41% with respect to l_{max} on data set B_0.7. On data set B_0.9, TALNS is superior to LNS with respect to *TT*; TALNS performs, on average, 1.59% better than LNS. Nevertheless, the solutions found by LNS have, on average, 7.48% smaller maximum arrival time differences.

The template-based approaches are suitable when service frequency is high. With high service frequencies the variation in the customer requirements and, therefore, in the daily routing plans is low. Accordingly, the template gives a good approximation of the daily routes. With low service frequencies, there are more customers that require only one service during the planning horizon. LNS inserts all customers into the routing plan first, and improves the time consistency in a second step. In this way, there is a greater

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

Table 3 Results for Data Set B_0.5

	TALNS			LNS			*TT* Imp (%)	l_{max} Imp (%)
Instances	*TT* (avg)	l_{max}	CPU (s)	*TT* (avg)	l_{max}	CPU (s)	to TALNS	to TALNS
Christofides_1_5_0.5	1,685.19	25.80	103.00	1,677.59	17.78	60.02	0.45	31.07
Christofides_2_5_0.5	2,619.03	16.01	52.70	2,597.02	12.20	44.48	0.84	23.79
Christofides_3_5_0.5	2,705.88	20.93	163.77	2,670.03	19.53	150.64	1.32	6.71
Christofides_4_5_0.5	3,543.11	16.83	157.30	3,376.09	13.26	148.27	4.71	21.18
Christofides_5_5_0.5	4,090.38	13.17	231.89	4,060.42	12.75	199.97	0.73	3.17
Christofides_6_5_0.5	2,878.42	34.71	106.76	2,891.35	30.70	30.96	−0.45	11.53
Christofides_7_5_0.5	4,764.38	24.84	40.47	4,678.29	18.73	33.18	1.81	24.61
Christofides_8_5_0.5	5,355.78	36.71	114.25	5,357.87	28.80	88.64	−0.04	21.55
Christofides_9_5_0.5	7,493.43	36.53	119.10	7,422.41	31.40	94.16	0.95	14.04
Christofides_10_5_0.5	9,628.98	28.19	228.58	9,402.92	27.31	175.03	2.35	3.13
Christofides_11_5_0.5	3,456.14	16.01	184.07	3,317.01	12.68	326.45	4.03	20.78
Christofides_12_5_0.5	2,905.22	9.32	108.89	2,894.34	7.89	91.62	0.37	15.33
Average	4,260.50	23.25	134.23	4,195.44	19.42	120.28	1.42	16.41

Table 4 Results for Data Set B_0.7

	TALNS			LNS			*TT* Imp (%)	l_{max} Imp (%)
Instances	*TT* (avg)	l_{max}	CPU (s)	*TT* (avg)	l_{max}	CPU (s)	to TALNS	to TALNS
Christofides_1_5_0.7	2,131.91	15.00	74.53	2,123.02	13.31	73.33	0.42	11.23
Christofides_2_5_0.7	3,615.43	15.43	47.85	3,599.69	13.81	54.79	0.44	10.52
Christofides_3_5_0.7	3,343.10	14.28	120.65	3,308.12	14.04	224.61	1.05	1.69
Christofides_4_5_0.7	4,691.71	10.50	199.10	4,562.46	9.76	245.18	2.75	7.09
Christofides_5_5_0.7	5,716.18	10.67	212.32	5,682.84	11.06	264.21	0.58	−3.61
Christofides_6_5_0.7	4,064.34	31.73	137.50	4,092.89	21.10	60.85	−0.70	33.51
Christofides_7_5_0.7	6,833.13	28.76	37.58	6,693.43	23.87	41.64	2.04	16.98
Christofides_8_5_0.7	7,225.29	27.39	141.94	7,192.44	27.31	132.23	0.45	0.31
Christofides_9_5_0.7	10,579.78	29.98	225.29	10,461.29	30.88	132.07	1.12	−2.99
Christofides_10_5_0.7	13,217.64	28.71	281.33	13,050.14	27.76	184.03	1.27	3.31
Christofides_11_5_0.7	4,772.98	6.40	177.24	4,810.15	6.53	495.98	−0.78	−2.04
Christofides_12_5_0.7	3,548.52	6.31	113.48	3,571.40	6.25	91.29	−0.65	0.90
Average	5,811.67	18.76	147.40	5,762.32	17.14	166.68	0.67	6.41

Table 5 Results for Data Set B_0.9

	TALNS			LNS			*TT* Imp (%)	l_{max} Imp (%)
Instances	*TT* (avg)	l_{max}	CPU (s)	*TT* (avg)	l_{max}	CPU (s)	to TALNS	to TALNS
Christofides_1_5_0.9	2,499.17	11.71	107.22	2,499.04	9.52	82.85	0.01	18.72
Christofides_2_5_0.9	4,045.92	9.07	148.99	4,051.71	8.88	64.91	−0.14	2.02
Christofides_3_5_0.9	4,018.88	9.49	135.69	4,035.72	8.56	281.54	−0.42	9.84
Christofides_4_5_0.9	5,027.71	7.97	267.20	5,049.91	7.37	375.27	−0.44	7.50
Christofides_5_5_0.9	6,623.67	3.88	427.53	6,882.86	3.96	390.74	−3.91	−2.07
Christofides_6_5_0.9	4,767.65	20.99	145.76	4,772.33	20.99	44.64	−0.10	0.00
Christofides_7_5_0.9	7,741.18	15.93	111.90	7,761.11	15.57	42.80	−0.26	2.27
Christofides_8_5_0.9	8,758.24	21.95	164.40	8,770.39	19.99	122.08	−0.14	8.95
Christofides_9_5_0.9	12,648.34	19.09	159.07	12,556.00	18.99	149.27	0.73	0.54
Christofides_10_5_0.9	16,076.38	16.14	282.28	16,146.57	16.52	218.61	−0.44	−2.37
Christofides_11_5_0.9	5,043.93	7.46	98.69	5,666.20	5.06	491.65	−12.34	32.19
Christofides_12_5_0.9	4,015.40	3.72	122.73	4,081.56	3.11	163.78	−1.65	16.34
Average	6,772.21	12.28	180.96	6,856.12	11.54	202.34	−1.59	7.83

Table 6 Results for Data Set C_small with Different Importance of Consistency

	One driver				Two drivers				Three drivers				Imp (%)
α	TT	l_{max}	OV	dev_{LNS} (%)	TT	l_{max}	OV	dev_{LNS} (%)	TT	l_{max}	OV	dev_{LNS} (%)	1 → 2 drivers
0.999	127.88	3.87	54.01	0.00	124.94	5.49	52.77	0.00	124.94	5.49	52.77	0.00	2.31
0.9	127.97	3.50	49.04	0.01	125.14	3.81	48.00	0.00	125.14	3.81	48.00	0.00	2.13
0.8	128.18	2.98	43.94	0.00	125.55	2.91	43.05	0.00	125.55	2.91	43.05	0.00	2.04
0.7	128.18	2.98	38.82	0.00	125.67	2.76	38.02	0.00	125.67	2.76	38.02	0.00	2.06
0.6	128.92	2.46	33.67	0.02	126.39	2.20	32.93	0.00	126.39	2.20	32.93	0.00	2.19
0.5	129.31	2.24	28.44	0.00	126.68	2.03	27.78	0.03	126.68	2.03	27.78	0.03	2.31
0.4	130.04	1.97	23.16	0.00	129.18	1.06	22.46	0.46	129.18	1.06	22.46	0.37	3.00
0.3	133.05	1.31	17.79	0.17	129.18	1.06	17.11	0.30	129.18	1.06	17.11	0.26	3.81
0.2	135.44	0.97	12.23	0.13	133.84	0.43	11.65	0.53	133.84	0.43	11.65	0.52	4.76
0.1	146.22	0.20	6.36	0.75	138.83	0.08	5.94	1.51	138.83	0.08	5.94	1.58	6.52
0.001	152.70	0.00	0.06	0.75	142.44	0.00	0.06	2.23	142.44	0.00	0.06	2.28	6.72

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

flexibility to place customers that are not linked to consistency requirements at positions that support the time consistency of frequent customers.

Flexible departure times improve arrival time consistency (Kovacs, Parragh, and Hartl 2014). The variation in each driver's departure times is modest; the average ratio between the standard deviation of the departure times and the shift length (among all instances in which the shift length is bounded) is 5.26%, 3.97%, and 2.39% for data sets B_0.5, B_0.7, and B_0.9, respectively. For example, if the shift is 10 hours, than a ratio of 5% indicates that each driver starts his route with a deviation of ±30 minutes from his average departure time.

5.4. Results for GenConVRP

In experiments on the GenConVRP, the focus is on examining the effect of the number of allowed drivers per customer (W) and the importance of time consistency (expressed by parameter α) on the results. We use a normalized objective function $f^n(s)$ in order to obtain similar dimensions for the total travel time and for the maximum arrival time difference regardless of the number of customers and the length of the planning horizon:

$$f^n(s) = \alpha \frac{10 \sum_{d\in D} \sum_{k\in K} \sum_{i\in N} \sum_{j\in N} t_{ij} x_{ijkd}}{\sum_{d\in D} \sum_{i\in N} w_{id}} + (1-\alpha) l_{\max}. \quad (31)$$

$f^n(s)$ sets 10 times the average travel time between two customers in relation to the maximum arrival time difference.

5.4.1. Comparison Between LNS and CPLEX on Data Set C_small. To verify the quality of the solutions obtained by LNS, we solve the small instances (C_small) and evaluate the gap to the solutions found by CPLEX. Table 6 shows the average results over all instances in data set C_small. The number of allowed drivers per customer is set to one, two, and three, respectively. OV denotes the average objective value over all instances. The results are obtained by CPLEX 12.5; all instances are solved to proven optimality. Columns dev_{LNS} give the average deviation in the objective value between LNS and CPLEX. The last column shows the improvement in OV that is obtained by allowing two drivers per customer instead of one.

Increasing the number of allowed drivers per customer from one to two leads to an average improvement of at least 2%. Allowing three drivers per customer has no effect on the solutions. So, better solutions are obtained by slightly increasing the number of drivers per customer, regardless of the decision maker's preference. Yet, allowing too many drivers per customer deteriorates driver consistency without improving the objective value.

The number of drivers also affects the relationship between the total travel time (TT) and the maximum arrival time difference (l_{max}). Decreasing α from 0.999 to 0.001, i.e., decreasing l_{max} by 100%, causes an increase of 19% in TT with one driver (see Table 6). With two and three drivers, TT increases by only 14%.

The LNS algorithm is able to find the best known solutions for the majority of the settings; the average

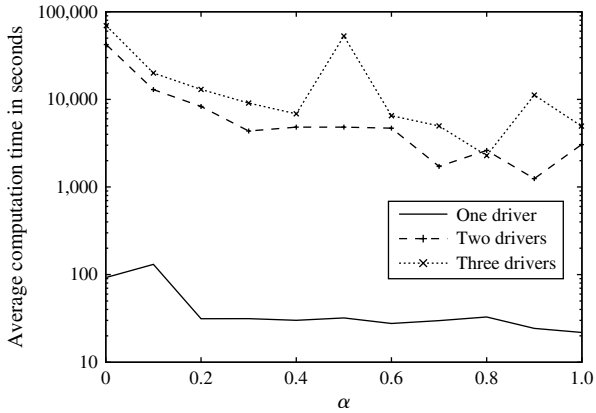


Figure 5 Average Computation Times Required by CPLEX

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

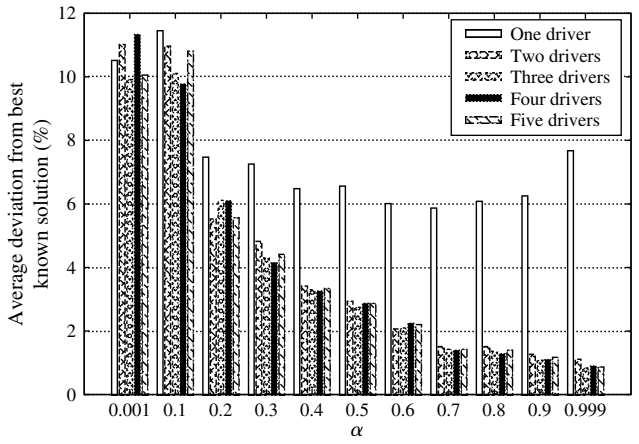


Figure 6 Average Deviation in the Objective Value for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.5

Note. The lower α, the more important the time consistency.

computation time is 1.7 seconds. LNS fails to find the highest quality solutions only when the focus is fully on time consistency, i.e., the optimal $l_{\max}$ is close to 0.

The average computation times required by CPLEX are shown in Figure 5 on a logarithmic scale. Multithreading was enabled in our experiments; the reported values refer to the total computation time across all threads. The computation times are given as a function of α and the number of allowed drivers W per customer. If W is one, the computation time increases from 21.9 seconds to 92.7 seconds as α decreases from 0.999 to 0.001; the peak is at $\alpha = 0.1$ with 130.9 seconds. With two drivers per customer, the average computation time is more than 11 hours at $\alpha = 0.001$. With three drivers per customer there are two peaks: one at $\alpha = 0.001$ and one at $\alpha = 0.5$ with an average computation time of 19.2 hours and 14.7 hours, respectively. CPLEX can cancel a large number of binary variables from the model if W is set to one; therefore, problems can be solved quickly. The computation time increases significantly when W is larger than one. The same observation is made by Francis, Smilowitz, and Tzur (2006). The authors enforce driver consistency in order to make the problem easier to solve with the proposed mathematical programming approach.

5.4.2. Results for Data Sets C_0.5, C_0.7, and C_0.9. In this section, we present results for large GenConVRP data sets C_0.5, C_0.7, and C_0.9. The results of all instances in the respective data set are aggregated for reasons of clarity and comprehensibility. Figures 6, 9, and 12 show the average improvement to the best known solution among all instances in data set C_0.5, C_0.7, and C_0.9, respectively. Results are presented for different parameters α and for different numbers of allowed drivers per customer. The best known solutions refer to the best results found with the respective setting for parameter α (regardless of the number of drivers per customer). For each α, the first bar represents the result for one driver per customer, the second bar the result for two drivers per customer, and so on.

The pattern in the figures is similar regardless of the service frequency. The LNS results have high variation with low α values. This result is not surprising because LNS is designed with a commercial perspective. Interestingly, with low α, better solutions are obtained if the driver consistency is more restrictive. This observation led to the improvement described in §4.7.

As α increases, the results are more stable and the advantage of allowing more than one driver per customer becomes apparent. For each service frequency, there is a very small difference between allowing two or more drivers per customer. However, in most of the cases the results improve significantly when two drivers are allowed instead of one. The benefit of allowing more than one driver per customer decreases with larger service frequency. This is because with higher service frequency the daily routes become similar. So,

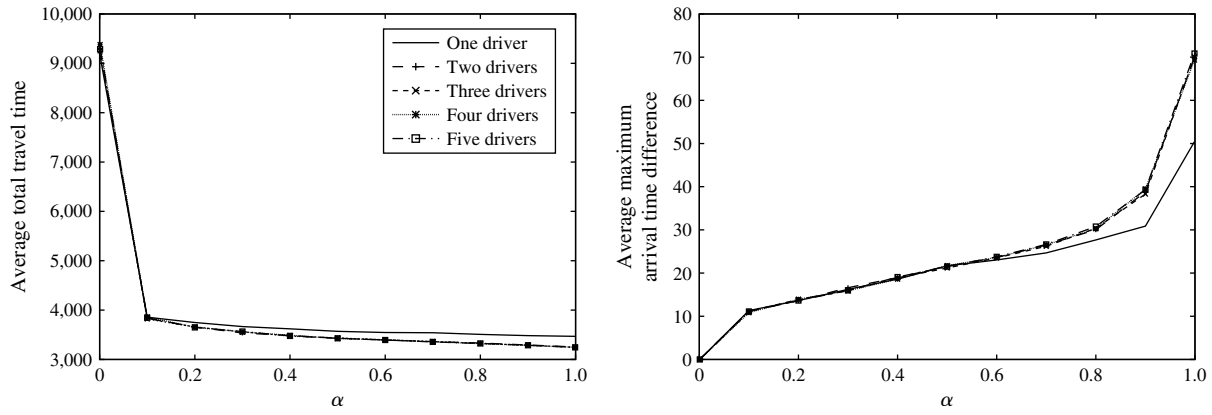


Figure 7 Average Total Travel Time and Average Maximum Arrival Time Difference for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.5

Note. The lower α, the more important the time consistency.

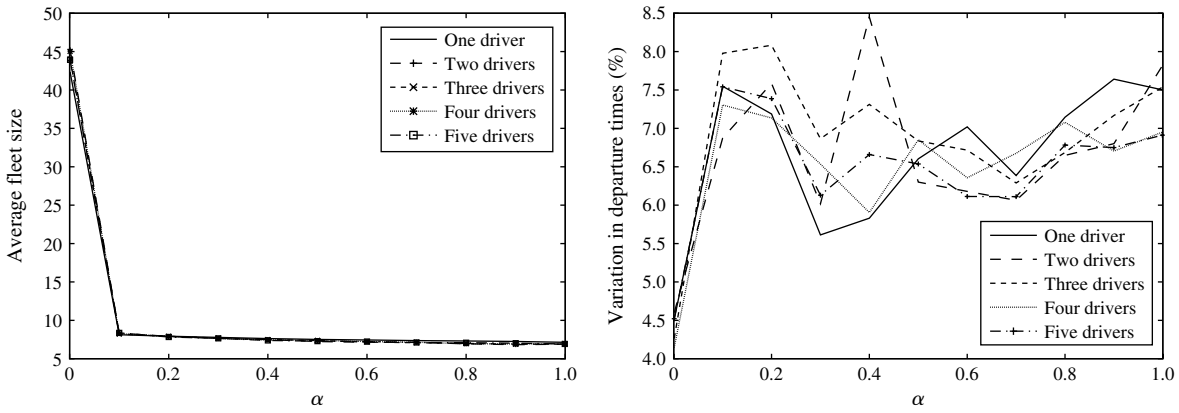


Figure 8 Average Fleet Size and Average Ratio Between the Standard Deviation of the Vehicle Departure Times and the Shift Length for Data Set C_0.5

Note. The lower α, the more important the time consistency.

not much can be saved by assigning a customer to different drivers on different days.

Figures 7, 10, and 13 give the average travel time and the maximum arrival time difference for data sets C_0.5, C_0.7, and C_0.9, respectively. The lines show the results with different numbers of allowed drivers per customer. When decreasing α from 0.999 to 0.9, there is a remarkable drop in the maximum arrival time difference. The total travel time, however, barely increases. With decreasing α, the maximum arrival time difference can be reduced further at the expense of a moderate increase in the total travel time. This observation is consistent with findings in Kovacs, Parragh, and Hartl (2014). LNS is able to generate solutions with a maximum arrival time difference of 0 if the focus is fully on time consistency ($\alpha = 0.001$). Yet, this comes at the price of a drastic increase in the total travel time.

Figures 8, 11, and 14 show the average fleet (or staff) size (i.e., the average number of vehicles used in the solutions) and the variation in each driver's departure times for data sets C_0.5, C_0.7, and C_0.9, respectively. The variation in the departure times is expressed by the average ratio between the standard deviation of the vehicle departure times and the shift length; we consider only instances in which the shift length is bounded.

The figures show that driver consistency has a minor impact on the average fleet size. For all tested service frequencies and parameters α, the average fleet size is almost the same regardless of the number of allowed drivers per customer. Time consistency can also be improved with a modest increase in the average fleet size. Yet, with $\alpha = 0.001$, an average fleet size that is 2.5 to six times larger (depending on the service frequency) than the fleet size without considering consistency is required. The need for a larger fleet explains the sharp increase in the total travel time in Figures 7, 10, and 13. The variation in the departure times is not affected much by the number of different drivers per customer and the emphasis on arrival time consistency. The ratios are roughly 7%, 5%, and 3% for data sets C_0.5, C_0.7, and C_0.9, respectively. These results are similar to the variations found in data sets B_0.5, B_0.7, and B_0.9 indicating that the effect of the time windows on the variation in the departure times is minor. The variations decrease only when $\alpha = 0.001$. Here, the number of routes increases but they become shorter and are, therefore, less affected by fluctuations in the demand.

5.4.3. Discussion of the Results. Except for instances where α is very small (which we assume are not interesting in commercial applications), we can summarize that the proposed LNS algorithm performs robustly as it generates high service quality solutions with different input parameters with a minor increase in the variable cost (i.e., travel time) and the fixed cost (i.e., fleet size). The results show that a very

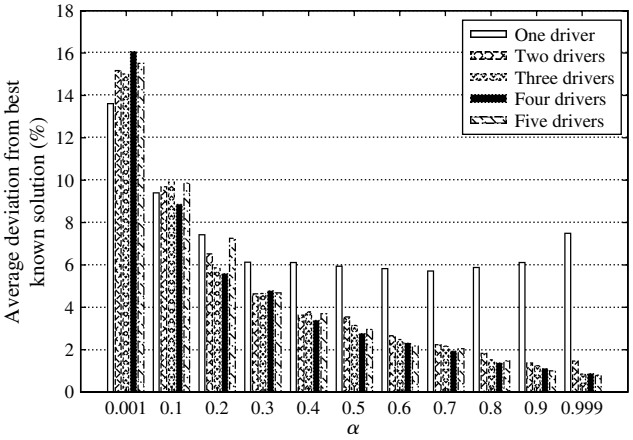


Figure 9 Average Deviation in the Objective Value for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.7

Note. The lower α, the more important the time consistency.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

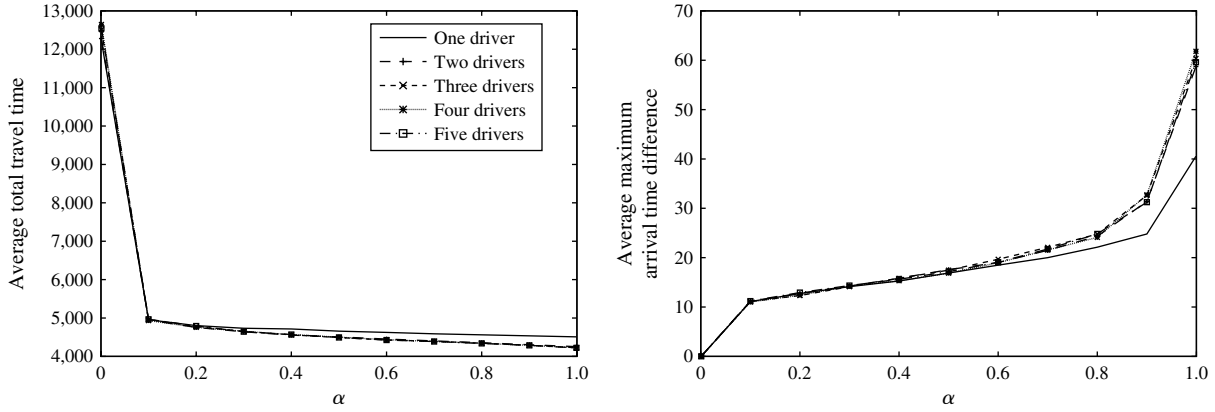


Figure 10 Average Total Travel Time and Average Maximum Arrival Time Difference for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.7

Note. The lower α, the more important the time consistency.

strict level of driver consistency with one driver per customer can be obtained without having to increase the fleet size. Yet, with greater flexibility, obtained by allowing at least two drivers per customer, the objective value can be reduced by up to 6.5% (see Figure 6 with $\alpha = 0.999$). The larger the irregularities in the customer demands, the greater the potential savings. Increasing time consistent service in order to improve customer satisfaction is recommended to service providers because a large reduction in $l_{\max}$ can be obtained with modest increases in the travel cost and the fleet size. The variation of the vehicle departure times necessary to improve arrival time consistency is slightly affected by time windows; the ratio between the standard deviation of the vehicle departure times and the shift length is between 3% and 7% in data set C (i.e., considering AM/PM time windows) and between 2.39% and 5.26% in data set B (i.e., without considering time windows).

Our findings are in line with the literature. In the ConVRP, Feillet et al. (2014) demonstrate cost savings of up to 7.5% if driver consistency is relaxed. In the inventory routing problem, visiting each customer by the same driver causes an increase in cost by up to 9.85% (Coelho, Cordeau, and Laporte 2012). Spliet

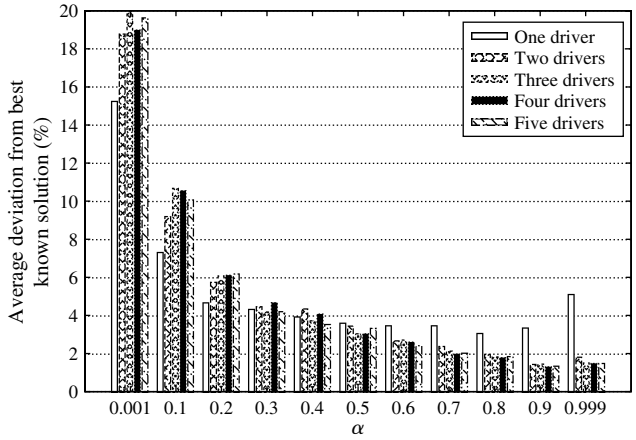


Figure 12 Average Deviation in the Objective Value for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.9

Note. The lower α, the more important the time consistency.

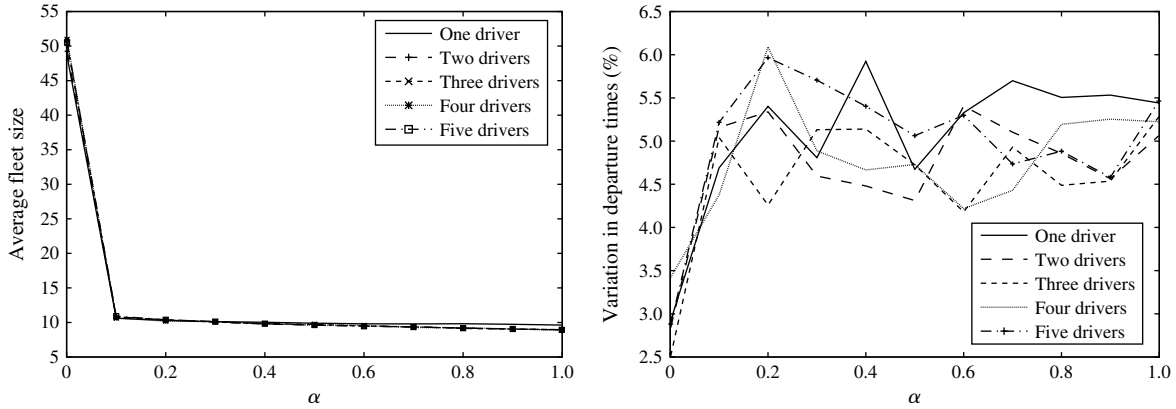


Figure 11 Average Fleet Size and Average Ratio Between the Standard Deviation of the Vehicle Departure Times and the Shift Length for Data Set C_0.7

Note. The lower α, the more important the time consistency.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

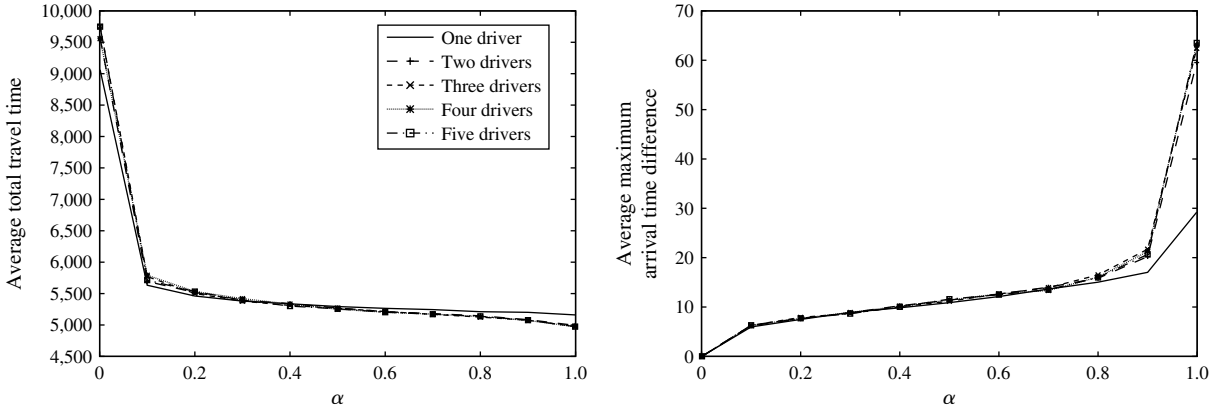


Figure 13 Average Total Travel Time and Average Maximum Arrival Time Difference for Different Numbers of Allowed Drivers per Customer and Different Importance of the Time Consistency on Data Set C_0.9

Note. The lower α, the more important the time consistency.

(2013) studies the driver assignment vehicle routing problem; the increase in cost decreases from 12% to 2.9% if a strict driver consistency is enforced only for 75% of the customers. Different variants of the periodic vehicle routing problem are investigated in Francis, Smilowitz, and Tzur (2007) and Smilowitz, Nowak, and Jiang (2013); improving driver consistency increases routing cost by up to 6.1% and 5.2%, respectively.

In Feillet et al. (2014) and Groër, Golden, and Wasil (2009), drivers have to start their route at time 0. With this restriction, the reported cost of improving arrival time consistency is 5.9% in Feillet et al. and between 6.6% and 15% in Groër, Golden, and Wasil. (The results in Groër, Golden, and Wasil also include the cost of visiting each customer by the same driver.) Kovacs, Parragh, and Hartl (2014) show that a 60% tighter constraint on $l_{\max}$ increases cost by up to 186.15% if departure times from the depot are fixed. However, with flexible departure times, arrival time consistency can be improved by increasing travel cost only between 0.63% and 1.62%, on average.

5.4.4. Effect of Time Windows. In this section, we examine the results with regard to the location of AM and PM customers, respectively. We compare three scenarios: First, the scenario described in §5.1, i.e., AM and PM customers are distributed homogeneously in the service region. In the second scenario, the depot is located in the city center and customers are assigned to time windows in a circular fashion; close business customers are visited in the morning and more remote private customers are visited in the afternoon. Time windows are ignored in the third scenario, i.e., each customer can be visited either in the morning or in the afternoon. Figure 15 illustrates the first and second scenario.

In Table 7, we present the interaction between parameters α, W, the service frequency, and the time window pattern. The values give the average percentage improvement in the objective value compared to scenario one for different α, W, and service frequency settings; "circular" refers to scenario two and "no TW" refers to the scenario in which time windows are

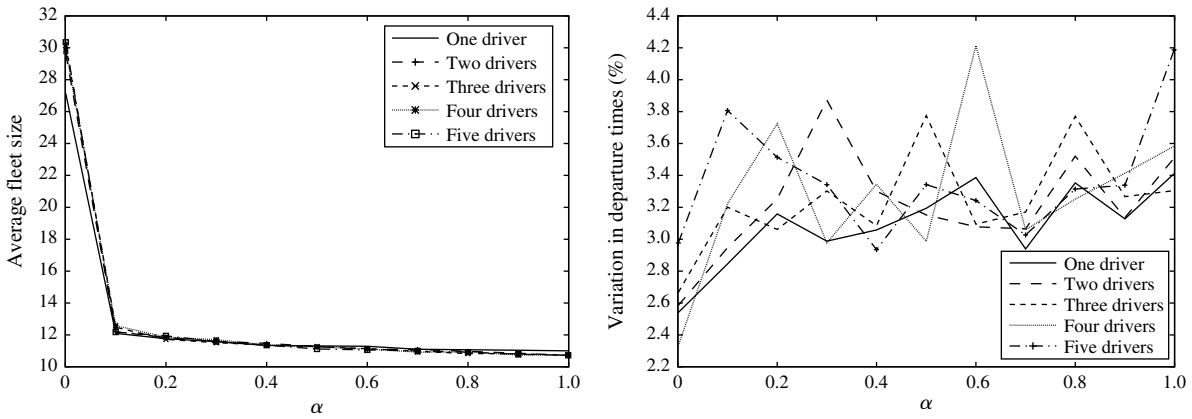


Figure 14 Average Fleet Size and Average Ratio Between the Standard Deviation of the Vehicle Departure Times and the Shift Length for Data Set C_0.9

Note. The lower α, the more important the time consistency.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

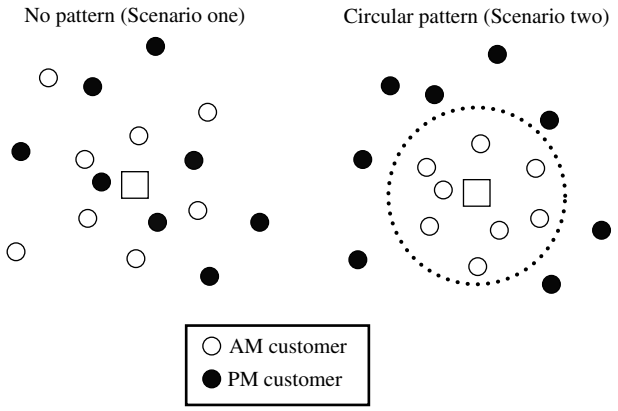


Figure 15 **Two Scenarios: First, There Is No Correlation Between Customer Location and Time Window Assignment; Second, Customers Are Assigned to Time Windows in a Circular Fashion**

ignored. The benefit of a circular pattern is between 0.68% and 6.13%. The actual improvement depends on the service frequency and parameter α; the level of driver consistency has a minor impact. Unattractive routes with crossing edges are inevitable when AM and PM customers are distributed homogeneously and α is small, i.e., the emphasis on arrival time consistency is large. However, routing can be performed efficiently when AM and PM customers are assigned in a circular fashion without deteriorating arrival time consistency; the resulting routing plans show a flower-shaped structure. Routing cost outweighs the importance of arrival time consistency as α increases; routes are optimized with regard to cost and detours in order to improve consistency are avoided. Therefore, the benefit of the circular time window assignment diminishes.

Small service frequencies, i.e., large demand fluctuations, make it difficult to plan routes efficiently in terms of cost and achieve high arrival time consistency at the same time. As described above, the circular time window pattern reduces the conflict between arrival time consistency and routing cost. So, the lower the service frequency the higher the improvement of having a circular time window pattern. We assume that exceptions from this statement occur because of variations in the heuristic results.

Ignoring time windows improves the results between 9.83% and 14.62%. The influence of the parameters is similar to the circular time window pattern. However, the impact of α is lower than before. Without time windows, the trivial upper bound for $l_{\max}$ increases from $T/2$ to T. Therefore, our algorithm struggles to improve the results when the emphasis is put on arrival time consistency.

Table 7 **Average Percentage Improvement in the Objective Value Compared to Scenario One (i.e., Time Windows Are Independent of the Location of Customers)**

		Service frequency					
		0.5		0.7		0.9	
W	α	Circular	No TW	Circular	No TW	Circular	No TW
1	0.5	6.13	14.62	4.02	11.36	3.16	10.66
	0.6	4.84	13.89	2.52	11.29	2.29	10.57
	0.7	4.44	13.38	1.94	11.14	2.53	10.92
	0.8	3.14	13.06	1.12	11.12	1.48	10.29
	0.9	2.71	12.86	0.68	10.89	1.24	10.45
2	0.5	5.71	14.23	3.80	11.31	3.65	10.97
	0.6	4.50	13.08	2.62	10.84	2.67	10.74
	0.7	3.91	13.31	1.96	10.42	2.14	10.53
	0.8	3.44	12.56	1.02	10.09	2.01	10.82
	0.9	2.81	12.18	1.17	9.83	1.28	10.16
3	0.5	5.65	14.02	3.27	11.23	3.04	10.75
	0.6	4.61	13.05	2.52	10.52	2.87	10.72
	0.7	3.85	12.52	2.03	10.58	2.26	10.75
	0.8	3.22	12.41	0.99	9.93	1.99	10.35
	0.9	2.94	12.27	1.23	10.08	1.37	10.45

Note. Circular refers to scenario two and no TW means that time windows are ignored.

6. Conclusion

In this paper, we presented an extension of the consistent vehicle routing problem. We call it the generalized consistent vehicle routing problem because it generalizes the ConVRP by allowing more than one driver per customer, by relaxing the constraint for the maximum arrival time difference, by allowing flexible vehicle departure times, and by incorporating AM/PM time windows for the customers. The GenConVRP is motivated by many real-world applications in which customer satisfaction can be increased by providing driver and time-consistent service. Examples are found in the small package shipping industry, in home health care, and in the transportation of elderly and handicapped people.

The GenConVRP is solved by a large neighborhood search algorithm (LNS). Many solution approaches for the ConVRP are based on the template concept in which daily vehicle routes are derived from a set of template routes. Our approach deviates from this principle and generates solutions without using templates. Experiments on different ConVRP variants show that on average the nontemplate-based LNS performs better than all published algorithms for the ConVRP.

We analyzed the effect of varying emphasis on time and driver consistency by generating and solving new test instances. Results show that high levels of time consistency can be provided with small increases in the travel cost. The variation of the vehicle departure times necessary to improve arrival time consistency is modest. In terms of driver consistency, a one-to-one driver-to-customer assignment might be too restrictive for commercial applications.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

An important managerial implication follows from the observation that allowing more than one driver per customer can reduce the routing cost by up to 6.5%. However, allowing more than two drivers per customer produces negligible cost savings. We also found that both driver and time consistency have a small impact on the fleet size, i.e., providing higher consistency requires only a minor increase in the staff size. Disregarding AM/PM time windows decreases cost by up to 14.62% when demand fluctuations are high and emphasis is put on service consistency.

The presented problem is defined for a fixed time interval. However, in many applications loyal customers remain with the same service provider for several years. Future research involves the integration of consistency requirements into infinite horizon models with changing customer demands. Another challenge for future research is to try to prove the optimality of Algorithm 2 or demonstrate a counterexample.

Acknowledgments

The authors would like to thank the two anonymous referees for constructive comments and suggestions. The fourth author is supported by the Austrian Science Fund (FWF): T514-N13. The computational results presented have partly been achieved using the Vienna Scientific Cluster. This support is gratefully acknowledged.

Appendix A. Destroy Operators

In this section, we describe the applied destroy operators in more detail.

Random removal. The random removal operator selects customers randomly and removes them from each day until u removals have been made. The purpose of this operator is to diversify the search process.

Related removal. The idea of the related removal operator is to remove customers that share similarities (Shaw 1998). This approach facilitates the interchange of customers and favors the grouping of similar customers on the same route. We define similarity between two customers i and j, $\mathcal{R}(i, j)$, in terms of geographical distance, difference in the demand, difference in the service time, and difference in the visit patterns (Ropke and Pisinger 2006b; Kovacs, Parragh, and Hartl 2014). Parameters λ, μ, ν, and ξ control the weight of each measurement.

$$\mathcal{R}(i, j) = \lambda t_{ij} + \mu|\bar{q}_i - \bar{q}_j| + \nu|\bar{s}_i - \bar{s}_j| - \xi\gamma_{ij}, \qquad \text{(A1)}$$

where $\bar{q}_i$ and $\bar{s}_i$ are the average demand and the average service time of customer i among all days he is visited, respectively; γ_{ij}, is a measure of similarity in terms of visit frequency (Kovacs, Parragh, and Hartl 2014). It is given by the number of days on which either both customers i and j are serviced or neither of them are serviced, i.e.,

$$\gamma_{ij} = |D| - \sum_{d\in D} |w_{id} - w_{jd}|. \qquad \text{(A2)}$$

The related removal operator starts by removing a randomly chosen customer from the solution and inserting him into the set of removed customers S. In each iteration, a customer is selected randomly from S and the similarity values between the chosen customer and all assigned customers are calculated. The customer with the highest similarity, i.e., lowest $\mathcal{R}(i, j)$, is removed from each day and added to set S. The process stops when at least u removals have been made.

Cluster removal. The cluster removal operator (Ropke and Pisinger 2006a) is applied on each day, one after another. The number of customers to remove on day d, u_d, is chosen from the interval $[\min\{0.1\sum_{i\in N\setminus\{0\}} w_{id}, 30\}, \min\{0.4\sum_{i\in N\setminus\{0\}} w_{id}, 60\}]$. On a given day, a route with at least three customers is selected randomly. The customers on that route are grouped into two geographical clusters by applying Kruskal's algorithm (Kruskal 1956) to find the minimum spanning tree among them.[4] We obtain two components that represent two clusters by deleting the longest edge in the tree. One of them is selected randomly and all customers in that cluster are removed. This approach is illustrated in Figure A.1.

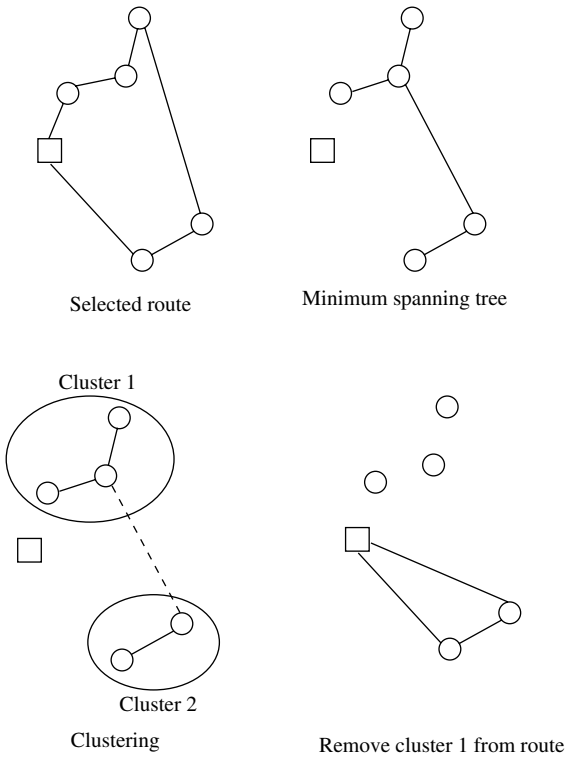


Figure A.1 Cluster Removal Operator

The next route to be destroyed is the one that contains the customer with the smallest distance to a randomly chosen customer from the previously removed cluster. The process of finding clusters and removing them continues until at least u_d customers have been removed or until no route with at least three customers is left. More than u_d customers may be removed in order to remove complete clusters.

Appendix B. Computing the Shifts in the Vehicle Departure Times

Algorithm 2 in §4.3 reduces the maximum arrival time difference by iteratively shifting the vehicle departure times. To

[4] The GenConVRP is defined on a digraph. We convert the digraph into an undirected graph by setting $t_{ij} = t_{ji} = \min\{t_{ij}, t_{ji}\}$.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

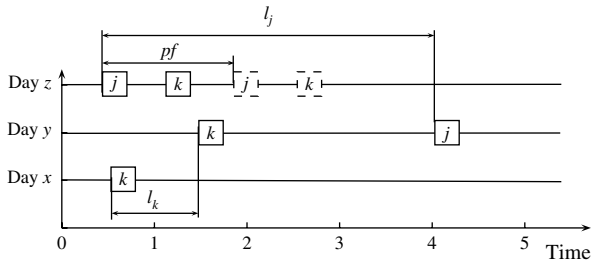


Figure B.1 Departure Time Adjustment: Scenario 1

compute the maximum amount of time customer j's starting time can be pushed forward with respect to customer k, $pf(j,k)$, we have to distinguish two scenarios.

In the first scenario, the day on which customer k is visited earliest is not the day on which customer j is visited earliest. This scenario is illustrated in Figure B.1. In this case we solve the following equation to get $pf(j,k)$:

$$l_j - pf(j,k) = l_k + pf(j,k) - (a_k^{\text{latest}} - a_k^{\text{current}}). \tag{B1}$$

Here, a_k^{current} denotes the arrival time at customer k on the current day and a_k^{latest} is the latest arrival time at customer k. The left-hand side is the arrival time difference of j and the right-hand side is the arrival time difference of k after delaying j's route on its earliest day by $pf(i,k)$. The push forward that leads to a maximum reduction in l_j with respect to customer k is

$$pf(j,k) = \frac{l_j - l_k + (a_k^{\text{latest}} - a_k^{\text{current}})}{2}. \tag{B2}$$

In the second scenario, the day on which customer k is visited earliest is the day on which customer j is visited earliest (see Figure B.2). We set $pf(j,k)$ by solving

$$l_j - pf(j,k) = (a_k^{\text{current}} + pf(j,k)) - a_k^{\text{2nd earliest}}, \tag{B3}$$

where $a_k^{\text{2nd earliest}}$ is the second earliest arrival time of customer k. The push forward that leads to a maximum reduction in l_j with respect to customer k is

$$pf(j,k) = \frac{l_j + a_k^{\text{2nd earliest}} - a_k^{\text{current}}}{2}. \tag{B4}$$

The approach of pulling routes backward follows the same principle as for pushing routes forward. Again, we have to distinguish two scenarios. First, customer k's arrival time is not the latest on the day j is visited latest. In this case the pull backward is

$$pb(j,k) = \frac{l_j - l_k + (a_k^{\text{current}} - a_k^{\text{2nd earliest}})}{2}. \tag{B5}$$

Otherwise, the pull backward is

$$pb(j,k) = \frac{l_j + a_k^{\text{current}} - a_k^{\text{2nd earliest}}}{2}. \tag{B6}$$

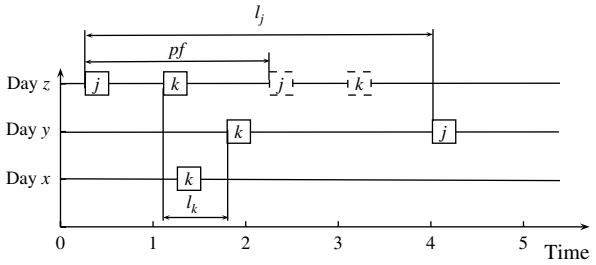


Figure B.2 Departure Time Adjustment: Scenario 2

Appendix C. Speeding Up the 2-Opt Heuristic

The 2-opt heuristic to improve time consistency in §4.4 is costly in terms of computation time. We can reduce the computational burden significantly by detecting unnecessary reversals in advance. If the resulting increase in the travel time (Δtt) would outweigh the resulting reduction in the maximum arrival time difference ($\Delta l_{\max}$), $\alpha \Delta tt > (1-\alpha)\Delta l_{\max}$, we can skip the current reversal. The computation of Δtt is fast because in the worst case it is linear in the number of customers on the route that is affected by the reversal. (The computation requires constant time if the distance matrix is symmetric.) Yet, $\Delta l_{\max}$ has to be estimated.

Proposition. *For each pair of customers (i,j) and each pair of days (x,y) where the customer pair is visited by the same driver, without taking the route on which* 2*-opt is to be performed into account, we can calculate a lower bound $\underline{\Delta l}_{\max}$ for $\Delta l_{\max}$ as follows*:

$$\underline{\Delta l}_{\max} = \max_{\substack{x,y \in D, x \neq y \\ i,j \in N \setminus \{0\}, i \neq j \\ k_{ix} = k_{jx}, x \neq d' \vee k_{ix} \neq k' \\ k_{iy} = k_{jy}, y \neq d' \vee k_{iy} \neq k'}} \frac{|a_{jy} - a_{iy}| - |a_{jx} - a_{ix}|}{2}. \tag{C1}$$

Parameter a_{ix} is the arrival time at customer i on day x and k_{ix} gives the corresponding driver. Route k' is the route and day d' is the day on which the 2*-opt heuristic is performed.*

Proof. Each pair of customers (i,j) that is visited on days x and y by the same driver may either be visited in the same order on both days (scenario 1 in Figure C.1) or in different order (scenario 2 in Figure C.1). In the first scenario, the arrival time difference that cannot be eliminated by shifting

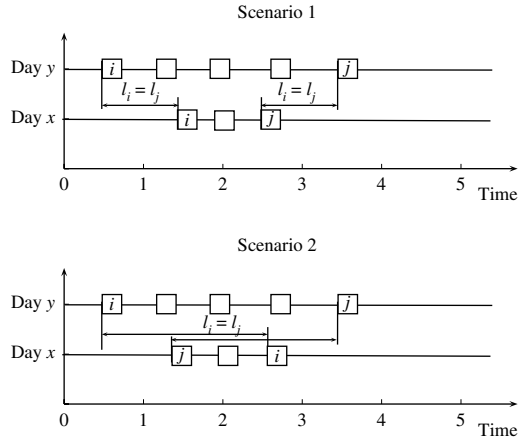


Figure C.1 Upper Bound on Maximum Arrival Time Difference

the departure times is $((a_{jy} - a_{iy}) - (a_{jx} - a_{ix}))/2$. Similarly, in scenario two, the maximum arrival time difference that cannot be eliminated by shifting is $(a_{jy} - a_{iy}) - ((a_{jy} - a_{iy}) - (a_{ix} - a_{jx}))/2$, which simplifies to $((a_{jy} - a_{iy}) - (a_{jx} - a_{ix}))/2$, yielding the same term as in scenario one. By taking the maximum value across all pairs of customers (i, j) and pairs of days (x, y), expression (C1) yields a valid lower bound for $l_{\max}$.

If the vehicle departure times are fixed to 0 (e.g., as in the original ConVRP) a lower bound for $\Delta l_{\max}$ is the maximum arrival time difference among all customers that are not affected by the reversal, i.e., are not visited on the route on which the 2-opt heuristic is applied.

References

Coelho LC, Laporte G (2013a) A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *Internat. J. Production Res.* 51(23–24):7156–7169.

Coelho LC, Laporte G (2013b) The exact solution of several classes of inventory-routing problems. *Comput. Oper. Res.* 40(2):558–565.

Coelho LC, Cordeau J-F, Laporte G (2012) Consistency in multi-vehicle inventory-routing. *Transportation Res. Part C: Emerging Tech.* 24(1):270–287.

Feillet D, Garaix T, Lehuédé F, Péton O, Quadri D (2014) A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks* 63(3):211–224.

Francis P, Smilowitz K, Tzur M (2006) The period vehicle routing problem with service choice. *Transportation Sci.* 40(4):439–454.

Francis P, Smilowitz K, Tzur M (2007) Flexibility and complexity in periodic distribution problems. *Naval Res. Logist.* 54(2):136–150.

Groër C, Golden B, Wasil E (2009) The consistent vehicle routing problem. *Manufacturing Service Oper. Management* 11(4):630–643.

Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680.

Kovacs AA, Parragh SN, Hartl RF (2014) A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks* 63(1):60–81.

Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.* 7(1):48–50.

Lei H, Laporte G, Guo B (2012) Districting for routing with stochastic customers. *EURO J. Transportation Logist.* 1(1–2):67–85.

Lin S (1965) Computer solutions of the traveling salesman problem. *Bell System Tech. J.* 44(10):2245–2269.

Paquette J, Bellavance F, Cordeau J-F, Laporte G (2012) Measuring quality of service in dial-a-ride operations: The case of a Canadian city. *Transportation* 39(3):539–564.

Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. *Comput. Oper. Res.* 34(8):2403–2435.

Potvin J-Y, Rousseau J-M (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur. J. Oper. Res.* 66(3):331–340.

Ropke S, Pisinger D (2006a) A unified heuristic for a large class of vehicle routing problems with backhauls. *Eur. J. Oper. Res.* 171(3):750–775.

Ropke S, Pisinger D (2006b) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Sci.* 40(4):455–472.

Schrimpf G, Schneider J, Stamm-Wilbrandt H, Dueck G (2000) Record breaking optimization results using the ruin and recreate principle. *J. Computational Phys.* 159(2):139–171.

Shaw P (1998) Using constraint programming and local search methods to solve vehicle routing problems. Maher M, Puget J-F, eds. *Principles and Practice of Constraint Programming CP98*, Vol. 1520 (Springer, Berlin Heidelberg), 417–431.

Smilowitz K, Nowak M, Jiang T (2013) Workforce management in periodic delivery operations. *Transportation Sci.* 47(2):214–230.

Spliet R (2013) Vehicle routing with uncertain demand. Ph.D. thesis, Erasmus University, Rotterdam, Netherlands.

Spliet R, Desaulniers G (2012a) The discrete time window assignment vehicle routing problem. Technical Report G201281, Les Cahiers du GERAD, Montreal.

Spliet R, Gabor AF (2012b) The time window assignment vehicle routing problem. Ei 2012-07, Econometric Institute, Erasmus School of Economics, Rotterdam, Netherlands.

Sungur I, Ren Y, Ordóñez F, Dessouky M, Zhong H (2010) A model and algorithm for the courier delivery problem with uncertainty. *Transportation Sci.* 44(2):193–205.

Tarantilis CD, Stavropoulou F, Repoussis PP (2012) A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems Appl.* 39(4):4233–4239.

Wong RT (2008) Vehicle routing for small package delivery and pickup services. Golden B, Raghavan S, Wasil E, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges*, Oper. Res./Comput. Sci. Interfaces Ser., Vol. 43 (Springer, New York), 475–485.

Woodward CA, Abelson J, Tedford S, Hutchison B (2004) What is important to continuity in home care? Perspectives of key stakeholders. *Soc. Sci. Medicine* 58(1):177–192.

Downloaded from informs.org by [216.73.216.187] on 14 July 2026, at 15:54 . For personal use only, all rights reserved.

This work is licensed under a Creative Commons Attribution 4.0 United States License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as "Transportation Science. Copyright 2015 INFORMS. http://dx.doi.org/10.1287/trsc.2014.0529, used under a Creative Commons Attribution License: http://creativecommons.org/licenses/by/4.0/us/"