



A note on the time travel approach for handling time windows in vehicle routing problems



Michael Schneider^{a,*}, Bastian Sand^a, Andreas Stenger^b

^a Chair of Business Information Systems & Operations Research, University Kaiserslautern, Germany

^b Lufthansa Technik Logistik Services GmbH, Hamburg, Germany

ARTICLE INFO

Available online 13 February 2013

Keywords:

Vehicle routing

Time windows

Efficient constraint handling

ABSTRACT

In this paper, we identify two cases in which the proposition for calculating time window penalties presented in Nagata, Y., Bräysy, O. and Dullaert, W. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows, *Computers & Operations Research* 2010;37(4): 724–37 yields incorrect results. We derive the corrected proposition and use numerical studies to show that a significant proportion of the evaluations performed by a Tabu Search for VRPTW falls under the two incorrect cases. Moreover, we demonstrate that the incorrect time window handling has a significant negative impact on the solution quality of the Tabu Search.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Nagata et al. [4] introduce a new, highly efficient approach to calculate time window violations, which is based on the concept of time travel. To put it short, if a time window violation occurs at a certain customer i , a penalty of the actual arrival time minus the latest feasible arrival time is assigned to the route. However, the calculation of violations for the following customers is executed as if a travel back in time to the latest feasible arrival time at customer i had taken place. In this way, none of the subsequent customer visits are affected by the actual delay at customer i .

This approach has two important advantages compared to the traditional approach of propagating time window violations along the route. First, it allows to evaluate potential time window violations for inter-route moves in constant time $\mathcal{O}(1)$ for most of the classical neighborhood operators like Relocate, Exchange, Or-Opt and 2-opt*. This significantly speeds up the calculation compared to the traditional approach that needs $\mathcal{O}(n)$ time. Second, the approach penalizes time window violations only at customers where they originate and feasible parts of routes are no longer penalized by earlier time window violations.

To keep this note short, we stick exactly to the notations and definitions used in the original paper and forgo unnecessary reintroductions. To compute time window violations, Proposition 1 [4, p. 736] states the following: If a route $r = \langle 0, \dots, x, v, y, \dots, 0 \rangle$ is constructed by inserting a node v in between two partial

paths $\langle 0, \dots, x \rangle$ and $\langle y, \dots, 0 \rangle$, the time window violation of the route is

$$TW(r) = TW_x^- + TW_y^+ + \max(\tilde{a}_x + s_x + c_{xv} - (\tilde{z}_y - c_{vy} - s_v), 0). \quad (1)$$

The presented proposition yields incorrect results under certain conditions. In the worst case, this leads to a situation where infeasible solutions are assigned a time window violation of 0. We identify the cases in which the given equation is wrong and present the corrected formula for each case (Section 2). Simple examples are used to illustrate the calculations. Furthermore, we perform numerical tests on standard benchmark instances of the Vehicle Routing Problem with Time Windows (VRPTW) in order to quantify the impact of the incorrect formula on the solution procedure of a classical single-solution metaheuristic like Tabu Search (TS) (Section 3). Finally, a brief conclusion is given in Section 4.

2. Derivation of correct proposition

We show that the presented proposition does not cover all cases concerning the time window characteristic of the customer v to be inserted. To this end, we distinguish three insertion cases:

1. Customer v is reached within its time window, i.e., we neither have to wait nor do we have to time travel backward.
2. Customer v is reached before the start of the associated time window, i.e., we have to wait before service can be started.
3. Customer v cannot be reached before the end of its time window, i.e., we have to time travel backward at v .

* Corresponding author. Tel.: +49 631 205 3765.

E-mail addresses: schneider@wiwi.uni-kl.de (M. Schneider),

sand@wiwi.uni-kl.de (B. Sand), stenger@wiwi.uni-frankfurt.de (A. Stenger).

Table 1

Time window handling related values for all customers in the example. Variables before insertion of the customers 3 and 4 into the initial route $\langle 0,1,2,0 \rangle$.

i	e	l	a	\tilde{a}	\tilde{a}'	\tilde{z}	\tilde{z}'	TW_i^+	TW_i^-
1	0	10	10	10	10	10	20	0	0
2	35	40	35	35	30	40	80	0	0
3	35	45	-	-	-	-	-	-	-
4	10	15	-	-	-	-	-	-	-

A simple example with customer set $I = \{1,2,3,4\}$ and a single depot 0 is used to show in which cases Eq. (1) fails. Let us assume the depot to have a scheduling horizon of $[0,100]$. All customers have a service time $s_i = 10, i \in I (s_0 = 0)$ and the distances $c_{ij}, i, j \in I \cup \{0\}, j \neq i$ are all 10. Travel times are assumed to be equivalent to traveled distances. In our example, we use the initial route $\langle 0,1,2,0 \rangle$ as starting point for all insertion cases. The remaining customers are individually inserted into the route between customers 1 and 2 to illustrate the cases where an error is produced. The time window handling related values for all customers are given in Table 1. In all cases, we compare the time window violation given by Eq. (1) to a stepwise calculation that has to walk through the complete newly created route. Contrary to the traditional time window handling, this stepwise calculation uses backward time travel to penalize violations in order to produce results that are comparable to those obtained by Eq. (1). For all insertion cases, we provide the corrected formula valid for the specific case and integrate these formulas into one equation that covers all cases afterwards.

2.1. Case 1: reach customer v within time window

If the customer v to be inserted is reached within its time window, Eq. (1) provides the correct result. The following step by step derivation illustrates this. Given the time window violation up to customer x (TW_x^+), we start service there at \tilde{a}_x and arrive at customer v at $\tilde{a}_x + s_x + c_{xv}$. We reach v within its time window and the arrival time at y can be calculated by adding $s_v + c_{vy}$. For the time window violation TW_y^- to be valid for completing the route from y onwards, we have to arrive at y before \tilde{z}_y . If the arrival time at y is later, we have to add the difference as time window violation.

Thus, if arrival at customer v is within the given time window, i.e., $e_v \leq \tilde{a}_x + s_x + c_{xv} \leq l_v$, the following holds (which is obviously equivalent to Eq. (1)):

$$TW(r) = TW_x^+ + TW_y^- + \max(\tilde{a}_x + s_x + c_{xv} + s_v + c_{vy} - \tilde{z}_y, 0).$$

2.2. Case 2: wait at customer v

If we have to wait to start service at the customer v to be inserted, Eq. (1) does not always provide the correct result as the example of route $r_2 = \langle 0,1,3,2,0 \rangle$ shows. The stepwise calculation is as follows: We arrive at customer 1 at time 10, leave at 20, arrive at customer 3 at 30, have to wait till 35, leave at 45 and arrive at customer 2 at 55. We have to time travel back 15 units, leave customer 2 at 50 and arrive at the depot at 60. The route r_2 has a time window violation of the amount we had to time travel, i.e., $TW(r_2) = 15$. According to Eq. (1), the violation is $TW(r_2) = 0 + 0 + \max(10 + 10 + 10 - (40 - 10 - 10), 0) = 10$, i.e., the amount of violation at y that is caused by the wait at v is not detected. Note that in the worst case, this means that the equation returns a time window violation of 0 for an infeasible route with a time window violation at customer y .

To correct the formula, we make the following considerations: At customer x with TW_x^+ , we start service at \tilde{a}_x and arrive at

customer v at $\tilde{a}_x + s_x + c_{xv}$. We reach v before its time window, wait until e_v and the arrival time at y can be calculated by adding $s_v + c_{vy}$ to e_v . As mentioned above, we have to arrive at y before \tilde{z}_y for the time window violation TW_y^- for completing the route from y onwards to be valid. If the arrival time at y is later, we add the difference as time window violation

$$TW(r) = TW_x^+ + TW_y^- + \max(e_v + s_v + c_{vy} - \tilde{z}_y, 0), \text{ if } \tilde{a}_x + s_x + c_{xv} < e_v. \tag{2}$$

Note that Eq. (1) is only incorrect for the considered case, if the wait at the inserted customer causes an arrival after the extended latest start time of the service at the following customer. Otherwise, it holds that $e_v + s_v + c_{vy} - \tilde{z}_y < 0$, and since $\tilde{a}_x + s_x + c_{xv} < e_v$, it follows that $\tilde{a}_x + s_x + c_{xv} - (\tilde{z}_y - c_{vy} - s_v) < 0$. Consequently, both Eqs. (1) and (2) can be simplified to $TW(r) = TW_x^+ + TW_y^-$ in this case.

2.3. Case 3: arrive late at customer v

The example route $r_3 = \langle 0,1,4,2,0 \rangle$ shows that Eq. (1) can also yield incorrect results if the time window at the inserted customer v is violated. The stepwise calculation is: We arrive at customer 1 at time 10, leave at 20, arrive at customer 4 at 30, have to time travel backwards 15 units, leave at 25 and arrive at customer 2 at 35, leave at 45 and arrive at the depot at 55. Consequently, route r_3 has a time window violation of $TW(r_3) = 15$. Eq. (1) returns an incorrect amount of violation of $TW(r_3) = 0 + 0 + \max(10 + 10 + 10 - (40 - 10 - 10), 0) = 10$. Note that again, in the worst case, a solution can be marked feasible according to Eq. (1) although the time window at customer v is violated.

The correct equation for this case can be derived as follows: At customer x with TW_x^+ , we start service at \tilde{a}_x and arrive at customer v at $\tilde{a}_x + s_x + c_{xv}$. We reach v after its time window, time travel back to l_v and add this as time window violation. The arrival time at y can be calculated by adding $s_v + c_{vy}$ to l_v . Thus, the following holds if we arrive late at customer v , i.e., if $\tilde{a}_x + s_x + c_{xv} > l_v$:

$$TW(r) = TW_x^+ + TW_y^- + (\tilde{a}_x + s_x + c_{xv} - l_v) + \max(l_v + s_v + c_{vy} - \tilde{z}_y, 0). \tag{3}$$

Note that Eq. (1) is only incorrect for the considered case if $l_v + s_v + c_{vy} - \tilde{z}_y < 0$, i.e., if we arrive before the extended latest start time of service at customer y . Otherwise, Eq. (3) is $TW(r) = TW_x^+ + TW_y^- + \tilde{a}_x + s_x + c_{xv} - l_v + l_v + s_v + c_{vy} - \tilde{z}_y$ and thus equal to Eq. (1).

2.4. Corrected proposition covering all cases

Summing up, the corrected proposition is: If a route $r = \langle 0, \dots, x, v, y, \dots, 0 \rangle$ is generated from two partial paths $\langle 0, \dots, x \rangle$ and $\langle y, \dots, 0 \rangle$ by inserting customer v , the time window penalty of this route is computed by

$$TW(r) = \begin{cases} TW_x^+ + TW_y^- + \max(\tilde{a}_x + s_x + c_{xv} + s_v + c_{vy} - \tilde{z}_y, 0) & \text{if } e_v \leq \tilde{a}_x + s_x + c_{xv} \leq l_v \\ TW_x^+ + TW_y^- + \max(e_v + s_v + c_{vy} - \tilde{z}_y, 0) & \text{if } \tilde{a}_x + s_x + c_{xv} < e_v \\ TW_x^+ + TW_y^- + (\tilde{a}_x + s_x + c_{xv} - l_v) + \max(l_v + s_v + c_{vy} - \tilde{z}_y, 0) & \text{if } \tilde{a}_x + s_x + c_{xv} > l_v. \end{cases}$$

This can be reduced to the following equation that covers all three cases:

$$TW(r) = TW_x^+ + TW_y^- + \max(\tilde{a}_x + s_x + c_{xv} - l_v, 0) + \max(\max(\min(\tilde{a}_x + s_x + c_{xv}, l_v), e_v) + s_v + c_{vy} - \tilde{z}_y, 0). \tag{4}$$

Download English Version:

<https://daneshyari.com/en/article/10348168>

Download Persian Version:

<https://daneshyari.com/article/10348168>

[Daneshyari.com](https://daneshyari.com)