Solving an On-line Capacitated Vehicle Routing Problem with Structured Time Windows

P. Hungerländer, K. Maier, J. Pöcher, A. Rendl and C. Truden*

12th December 2016

Abstract

The capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) is concerned with finding optimal tours for vehicles with given capacity constraints to deliver goods to customers within assigned time windows that can hold several customers and have a special structure (in our case equidistant and non-overlapping).

In this work, we consider an on-line variant of the cVRPsTW that arises in the online shopping service of an international supermarket chain: customers choose a delivery time window for their order online, and the fleet's tours are updated accordingly in real time. This leads to two challenges. First, the new customers need to be inserted at a suitable place in one of the existing tours. Second, the new customers have to be inserted in real time due to very high request rates. This is why we apply a computationally cheap, two-step approach consisting of an insertion step and an improvement step. In this context, we present a Mixed-Integer Linear Program (MILP) and a heuristic that employs the MILP. In an experimental evaluation, we demonstrate the efficiency of our approaches on a variety of benchmark sets.

Keywords: Vehicle routing problem; time windows; online optimization; exploiting special structure; mixed-integer linear programming.

1 Introduction

The online market has been a growing sector for decades, and customers are increasingly interested in doing their weekly grocery shopping through the internet. This is why all main supermarket chains now provide online delivery services, where customers can select goods on the internet that are then delivered to their homes within a time window that the customer selects.

Online ordering poses new challenges to the grocery suppliers, since the customers select the delivery time window, and not the supplier. This makes organising the delivery fleet more difficult and leads suppliers to build their delivery schedule in an on-line

*Corresponding Author: Department of Mathematics, Alpen-Adria Universität Klagenfurt, Austria, christian.truden@aau.at.

fashion, where the tours/schedules of the delivery vans are updated as new customer orders come in. Moreover, all these steps have to be performed as quickly as possible (within milliseconds) to provide a prompt online service to the customers.

In this paper, we tackle this problem in the context of a large international supermarket chain. The process of updating the vans' schedules is performed in recurring two steps, for each order a customer places online.

Insertion step: The customer receives a selection of available time windows from which he can choose one for delivery. The larger the selection, the more satisfied the customer is with the service. The customer selects one of the available time windows and is accordingly inserted into the current delivery schedule.

Improvement step: After the customer has successfully selected a time window, the system improves the current schedule by applying an improvement step. This step is essential to find as many feasible time windows as possible for the following customers and of course also to schedule as many customers as possible in total.

Within both above steps lies an optimization problem. The first one is concerned with inserting a customer optimally into an existing schedule, computing all time windows at which a given customer can be feasibly inserted. The second optimization problem is concerned with optimizing the existing, incomplete schedule, where the objective is to minimize the fleet's travel time without moving customers from their assigned time window. We denote this problem as the capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW).

For a recent, very good survey of the cVRPTW we refer to [1]. Toth and Vigo [5] give an overview over several types of the VRP including an extensive overview of different heuristics, integer programming approaches and case studies. Yang et al. [6] propose a closely related method to our simple insertion heuristic. Campbell and Savelsbergh [2] define the Home Delivery Problem (HDP), which is based on a similar application as our use case. However, they do not exploit the special time window structure and consider a different objective function. Finally, Ioannou et al. [3] define a similar real world problem, but for the traditional VRPTW.

In this short paper, we present a Mixed-Integer Linear Program (MILP) and a heuristic approach that deal both with the insertion and the improvement steps arising within our on-line variant of the cVRPsTW. Furthermore, we present preliminary experimental results on some benchmark instances.

This paper is organized as follows. In Section 2, we give a problem description, in Section 3 we outline approaches for inserting new customers into an existing schedule, and in Section 4 we present approaches for optimizing an incomplete schedule. In Section 5 we present computational results and Section 6 concludes the paper.

2 Problem Description

The capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) arises when delivering goods to customers who choose the delivery time window. The main difference to classical variants of the VRP are that the customer chooses the time window for delivery, and that the time windows have a special structure (in our case equidistant and non-overlapping) that can be computationally exploited. We are given:

1. customers $a_i,\ i \in \mathcal{C}$, with assigned weights w and service times s,
2. a set of time windows $\mathcal{W}$, $|\mathcal{W}| = t$,
3. travel times for each pair of customers and
4. a finite set of tours $\mathcal{S} := \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots\}$, $|\mathcal{S}| = m$, with assigned capacities $C_i,\ i \in \mathcal{S}$.

The aim of the cVRPsTW is to find a feasible schedule with minimal travel time.

3 Inserting New Customers

In our application, the customer places an order online, and the system proposes time windows during which the order can be delivered to the customer, who then selects a time window. In this section we outline our approach for finding this set of time windows, for which we use a simple insertion method, that takes a customer $\tilde{a}$, a tour $\mathcal{A}$ and a time window ω, and tries to insert $\tilde{a}$ into $\mathcal{A}$ at time window ω. Therefore, to calculate the set of available time windows for a new customer $\tilde{a}$, we apply the heuristic for each time window $\omega \in \mathcal{W}$ to each tour $\mathcal{A}$ in the schedule. Once the customer has selected a time window, we apply the simple insertion method again and choose from all feasible insertion points of the selected time window the one that leads to the smallest increase in travel time.

Facilitating calculations via earliest/latest arrival times. To faciliate the calculation of the feasible insertion points, we define the notion of an earliest and latest arrival time for each customer on a tour. It corresponds to the earliest, respectively the latest time, at which the van may arrive at a customer within time window ω, respecting the time windows of all other customers on the tour. When inserting customer $\tilde{a}$ between customers a_i and a_{i+1} we calculate the earliest and latest arrival time for $\tilde{a}$. These values are solely depended on the earliest, respectively the latest arrival time of the previous customer a_i and the following customer a_{i+1}, and time window ω.

Using the earliest and latest arrival time, a simple condition suffices to check if there is enough time between customer a_i and a_{i+1} to insert $\tilde{a}$ such that all customer orders can still be delivered within their assigned time windows. The condition is the following: customer $\tilde{a}$ can be inserted into a given tour between customer a_i and a_{i+1} in ω, if and only if the earliest arrival time of $\tilde{a}$ is less or equal than the latest arrival time of $\tilde{a}$. This condition allows to precompute the vast majority of the calculations that are needed to decide if the insertion results in a feasible tour.

Extending the heuristic with a MILP. The heuristic does not change the order of existing customers on the tour when inserting a new customer, therefore the insertion is not very advanced. In order to perform more sophisticated insertion operations, we utilize a MILP once the heuristic cannot find more feasible insertions.

We solve the Traveling Salesman Problems with structured Time Windows (TSPsTW) that is concerned with minimizing the travel times of a single tour $\mathcal{A}$ of the cVRPsTW. Hence in our setup, all tours in schedule $\mathcal{S}$, except $\mathcal{A}$, are fixed, and we solve the TSPsTW as feasibility problem (without objective function). In addition to the notation from Section 2, we use the following parameters and variables:

- a_1 is the start depot, a_n is the final depot and $\{a_2,\ldots,a_{n-1}\}$ is the set of customers assigned to tour $\mathcal{A}$.
- $[n_i],\ i\in[t]$, are the customers assigned to time window i.
- $t_{ij},\ i\in[n_k], j\in[n_\ell], k,\ell\in[t], i\neq j, k\leq\ell\leq k+1$, is travel time from customer a_i to customer a_j plus service time at customer a_i.
- $b_i,\ i\in[t]$, and $e_i,\ i\in[t]$, are the start and end time of time window i respectively.
- $z_i\in\mathbb{R}^+,\ i\in[t]$, gives the wait time during time window i.
- $x_{ij}\in\{0,1\},\ i\in[n_k], j\in[n_\ell], k,\ell\in[t], i\neq j, k\leq\ell\leq k+1$, with the interpretation:

$$x_{ij}=\begin{cases}1, & \text{if customer } j \text{ is visited right after customer } i,\\ 0, & \text{otherwise.}\end{cases}$$

The following constraints have to be satisfied after inserting a new customer into a given tour within a given time window:

$$\sum_{\substack{j\in[n_\ell],\ell\in[t]\\ j\neq i,\ k\leq\ell\leq k+1}} x_{ij}=1, \qquad i\in[n_k],\ k\in[t],\ i\neq n, \tag{1}$$

$$\sum_{\substack{i\in[n_k],\ k\in[t]\\ i\neq j,\ k\leq\ell\leq k+1}} x_{ij}=1, \qquad j\in[n_\ell],\ \ell\in[t],\ j\neq 1, \tag{2}$$

$$\sum_{\substack{i,j\in S,\\ i\neq j}} x_{ij}\geq|S|-1, \quad \forall S\subset[n_k]\setminus\{s,f\},\ k\in[t],\ |S|\geq 2, \tag{3}$$

$$\sum_{\substack{i\in[n_k],\ j\in[n_\ell]\\ k\leq\ell\leq k+1,\ k<h,\ i\neq j}} t_{ij}x_{ij}+\sum_{i<h} z_i\geq b_h, \qquad h\in[t]\setminus\{1\}, \tag{4}$$

$$\sum_{\substack{i\in[n_k],\ j\in[n_\ell]\\ k\leq\ell\leq k+1,\ k,\ell\leq h,\ i\neq j}} t_{ij}x_{ij}+\sum_{i\leq h} z_i\leq e_h, \qquad h\in[t], \tag{5}$$

$$x_{ij}\in\{0,1\}, \qquad i\in[n_k],\ j\in[n_\ell],\ k,l\in[t],\ i\neq j,\ k\leq\ell\leq k+1, \tag{6}$$

$$z_i\geq 0, \qquad i\in[t]. \tag{7}$$

Equalities (1) and (2) ensure that all vertices except the final depot a_n have exactly one outgoing edge and all vertices except the start depot a_1 have exactly one ingoing edge. Inequalities (3) are the subtour elimination constraints that we do not add directly to our MILP but handle through separation. Finally inequalities (4) and (5) guarantee that the arrival time of all customers is not before the start and not after the end of their assigned time window.

The most similar MILP formulation to our model is presented in [4], where, however, each time window contains only one customer. This is a critical difference to our version of the TSPTW, where several customers fit into a single time window, which we exploit in our approaches.

4 Optimizing Tours

The improvement step follows the insertion step, and is applied on the tour $\mathcal{A}$ into which the new customer has been inserted. We add the following objective function to the MILP above in order to minimize the travel time and hence increase the chances to insert further customers into $\mathcal{A}$ at a later point :

$$\min \sum_{\substack{i\in[n_k],\ j\in[n_\ell],\ k,\ell\in[t] \\ k\leq\ell\leq k+1,\ i\neq j}} t_{ij}x_{ij}. \tag{8}$$

5 Computational Experiments

In this section we present computational results. All experiments were performed on a Windows 7 64-bit machine equipped with an Intel Core i5-5300U (2×2300 MHz) and 12 GB RAM in single processor mode. We use Gurobi 6.5.1 as an IP-solver.

Benchmark instances. Our benchmark instances consist of customers where the coordinates on a square-grid are sampled from a two-dimensional uniform distribution and the travel times are calculated as the Euclidean distance between customers rounded to integers. Customer weights are sampled from a truncated normal distribution with mean of 7 and standard deviation of 2. Each customer is randomly assigned to one of the equidistant time windows. In our experiments we use instances, denoted e.g. as C100t7c150w5, consisting of 100/200/300 customers and 7 tours that have a capacity of 150/300/450 and 5/10/15 time windows each. Due to the size of the time windows and length of the service times of the customers there cannot be more than 6 customers within a time window per tour. The benchmark instances are designed to reflect real-world problems that arise in online shopping and can be downloaded from `http://tinyurl.com/vrpstw`.

Benchmark process. We iteratively insert new customers into the schedule, simulating customers placing orders online, where the preferred time window of the customer corresponds to the time window stated in the benchmark instance. Additionally we track how many time windows are available to each customer. We evaluate how many customers we are able to schedule as well as the runtime. Furthermore, we determine how many additional feasible time windows can be found when applying the MILP. For the improvement step we measure the improvement of the schedule compared to the schedule directly after each insertion step.

Our results summarized in Table 1 show that all our approaches use very little time (as required) and provide satisfactory results with respect to solution quality: we are able to determine insertion points for most time windows via our simple insertion heuristic and the MILP yields further improvements in both steps.

Approach	per step	C100t7c150w5	C200t7c300w10	C300t7c450w15
Heuristic-**insertion**	av. no. of windows	4.884	9.858	14.835
	av. time	0.17 ms	0.23 ms	0.23 ms
MILP-**insertion**	av. no. of add. windows	0.016	0.023	0.046
	av. time	8.23 ms	23.64 ms	49.98 ms
MILP-**optimization**	av. impr. over **insert. step**	0.158 %	0.096 %	0.061 %
	av. time	7.49 ms	13.56 ms	21.51 ms

Table 1: We state average values over five benchmark instances each. On the one hand we compare insertion via our simple insertion heuristic and via our MILP feasibility problem. On the other hand we present the average improvement of the objective function per iteration obtained by our MILP optimization problem.

6 Conclusion

In this paper we presented the capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) that arises in the context of delivering goods, where customers choose the delivery time window, and the delivery schedule is updated as new customers arrive. We introduced a two-step approach where we employ a heuristic and a Mixed-Integer Linear Program (MILP) in the respective insertion and improvement step. Our computational evaluation demonstrates that our approaches comply with the strict time limits and can produce good results within milliseconds, rendering them applicable to a real-world setting. For future research it would be interesting to extend our approach to further, more advanced heuristics and a MILP that improves the delivery schedule for a specific time window over all tours.

References

[1] R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6, 2012.

[2] A. M. Campbell and M. W. P. Savelsbergh. Decision support for consumer direct grocery initiatives. *Transportation Science*, 39(3):313–327, 2005.

[3] G. Ioannou, G. Kritikos, and G. Prastacos. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *The Journal of the Operational Research Society*, 52(5):523–537, 2001.

[4] A. Lodi and M. Milano. A hybrid exact algorithm for the tsptw. *INFORMS Journal on Computing*, 14:403–417, 2002.

[5] P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.

[6] X. Yang, A. K. Strauss, C. S. M. Currie, and R. Eglese. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation Science*, 2014.