Computers & Industrial Engineering 76 (2014) 366–377

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

An exact solution for vehicle routing problems with semi-hard resource constraints ☆

Khaled S. Abdallah [a,*], Jaejin Jang [b]

[a] Department of Supply Chain Management, College of International Transport and Logistics, Arab Academy for Science and Technology, Cairo, Egypt
[b] Industrial & Manufacturing Engineering Department, College of Engineering and Applied Science, University of Wisconsin-Milwaukee, Milwaukee, United States

ARTICLE INFO

Article history:
Received 30 November 2013
Received in revised form 4 July 2014
Accepted 15 August 2014
Available online 4 September 2014

Keywords:
VRPTW
Exact solution
Column generation
Semi-hard time windows

ABSTRACT

This paper presents an exact solution procedure for a vehicle routing problem with semi-hard resource constraints where each resource requirement can be relaxed to a pre-fixed extent at a predefined cost. This model is particularly useful for a supply chain coordination when a given number of vehicles cannot feasibly serve all the customers without relaxing some constraints.

It is different from VRP with soft time windows in that the violation is restricted to a certain upper bound, the penalty cost is flat, and the number of relaxations allowed has an upper bound.

We develop an exact approach to solve the problem. We use the branch cut and price procedure to solve the problem modeling the pricing problem as an elementary shortest path problem with semi hard resource constraints. The modeling of the subproblem provides a tight lower bound to reduce the computation time. We solve this subproblem using a label setting algorithm, in which we form the labels in a compact way to facilitate incorporation of the resources requirement relaxation information into it, develop extension rules that generate labels with possible relaxations, and develop dominance criteria that reduce the computation time. The lower bound is improved by applying the subset-row inequalities.

1. Introduction

Scheduling and dispatching of vehicles is one of the main activities in distribution and dispatching. This logistics problem is getting more important as more customers order remotely. This paper extends the traditional Vehicle Routing Problem with Time Windows (VRPTW) by allowing the resource windows to be semi-hard. A Semi-hard window can be relaxed to a predetermined extent at a predetermined compensation cost. It is soft in that it can be relaxed with a compensation cost, and hard in that the relaxation cannot exceed a certain limit. The compensation cost is fixed regardless of the extent a window is relaxed within the semi hard window. This model is useful when a delivery system cannot meet the customers' requirements without violating the original resource windows, and/or when the relaxation of resource windows can be pre-arranged for a supply chain coordination to reduce the total delivery and compensation cost.

This paper presents an exact solution of the problem, which is *NP*-hard, based on the column generation (CG) method. The CG method divides the VRP into a simple master problem with a restricted set of possible routes and a subproblem that keeps updating that restricted set. The subproblem generates a set of routes, each of which can be feasibly served by a single vehicle. Then, the master problem assigns a selected set of routes to vehicles to get a new solution of the problem dropping the integral constraints. CG keeps iterating between the master problem and the subproblem until no more routes are generated by the subproblem, or until meeting certain exit criteria. If the current solution of the iteration is not feasible to the original problem (i.e., it is not integral), then the CG method branches on selected criteria. The algorithm keeps iterating then branching until finding an optimal solution. The application of the CG method to the VRPTW is tracked back to Desrosiers, Soumis, and Desrochers (1984). Dumas, Desrosiers, and Soumis (1991) extend the problem to pickup-and-delivery with time window. Desrochers, Desrosiers, and Solomon (1992) formulates the master problems of a few variations of the VRPTW as set covering problems and the subproblems as non-elementary shortest path problems with resource constraints.

In the CG method, the subproblem is already *NP* hard, so various types of relaxations of its constraints are suggested. Kohl, Desrosier, Madsen, Solomon, and Soumis (1999) relax the subproblem by allowing routes with no more than k cycles. Feillet, Dejax, Gendreau, and Gueguen (2004) prevent cyclic paths and solve an elementary shortest path problem with resource constraints. They

☆ This manuscript was processed by Area Editor I-Lin Wang.
* Corresponding author. Tel.: +20 1281115220.
E-mail addresses: Khaled.Seif@aast.edu (K.S. Abdallah), Jang@uwm.edu (J. Jang).

http://dx.doi.org/10.1016/j.cie.2014.08.011
0360-8352/

extend the label setting algorithm of Beasley and Christofides (1989) by adding a binary vector that tells whether a customer is reachable from the current customer without violating its time window and demand requirement. This vector makes the search process to be active instead of being passive. Chabrier (2005) also solves the subproblem using the elementary shortest path formulation finding better lower bounds than those found with non-elementary paths. Irnich and Villeneuve (2006) solve the subproblem using a non-elementary shortest path algorithm with the elimination of cycles of size 3 or more. Righini and Salani (2006) solve the subproblem as elementary shortest path problem with resource constraints by generating search labels in forward and backward directions. This reduces the number of labels generated when accelerating techniques are used. They later develop a faster algorithm (Righini & Salani, 2008) by allowing the generation of routes with cycles for some customers, and then customers frequently included in the cycles are not allowed to cycle through later iterations.

The CG method can generate many routes at each iteration without significantly improving the solution; hence it is helpful to add cuts to the master problem after a column generation iteration ends to improve the lower bound. A cut is an inequality that is satisfied by all feasible solutions but currently it is not a part of the current formulation. When a cut is added to the master problem, it helps in separating some of the non optimal solutions. This changes CG into branch cut and price. It is helpful to add cuts to tighten the lower bound of the master problem. Fukasawa et al. (2006) tighten the master problem by adding several types of inequalities to the capacitated vehicle routing problem. The separation problems are already *NP* hard, so the time consumed in generating cuts and the quality improvement of the bound should be balanced. Jepsen, Petersen, Spoorendonk, and Pisinger (2008) successfully apply the subset row inequalities inspired by the clique inequality and the odd-hole inequality to the VRPTW. Peterson, Pisinger, and Spoorendonk (2008) apply the Chavatal–Gomory rank 1 cuts to solve VRPTW to tighten the lower bound of the master problem. They reduce the integrity gap at the root node for several instances, but the separation problem is more expensive than the subset row inequalities.

This paper introduces an exact solution procedure for the VRP with semi-hard resource constraints. The problem is solved using the branch cut and price algorithm. In Section 2, we formulate the problem and decompose this formulation into a master problem and a subproblem as an elementary shortest path problem with semi-hard resource constraints. In Section 3, we introduce an algorithm to solve the subproblem. In Section 4, we present numerical results. Finally, in Section 5 we summarize our findings and our future research plan.

2. Problem formulation

2.1. Problem description

We consider a problem where commodity needs to be delivered to customers meeting their resource windows. The windows are semi-hard as was explained above. There are multiple vehicles with the same capacity. The travel times and distances between the customers are known. The consumption of each resource is known at each customer. A customer can be visited only once. All the customers' requirements have to be met possibly after relaxing some of the resource windows. The objective is to minimize the delivery cost and the compensation cost of the relaxation.

2.2. Window and resource types

This model considers various types of resources and their windows; one of them is the traditional time window. There are two types of windows. The first type is the fixed type window, where the amount of consumption of a resource until a certain point cannot be decreased. This includes time window, arc length window, and total route length window. The other type is the adjustable type window, where the amount of consumption of a resource until a certain point can be decreased. This includes demand window and the number of long tours window.

A *time window* of a customer specifies the earliest and latest times the customer wants the service or delivery to start, and has the form of $[a,b]$, where a and b are the lower and upper limits of the window, respectively. If a time window is relaxed by δ, it becomes either $[a-\delta,b]$ or $[a,b+\delta]$. An *arc-length window* of a customer specifies the minimum and maximum length of any one-way travel to a customer. This window is used when there is a minimum compensation for short trips and/or fixed amount of overtime compensation for long trips. After relaxation, the window can be either $[a-\delta,b]$ or $[a,b+\delta]$. A *total route-length window* specifies the maximum distance (or time) of an entire route. When the total route-length is greater than the upper bound, a fixed compensation cost is charged. After relaxation, it becomes $[0,b+\delta]$.

The above three window types are fixed type because the service time at each node (for the time window), the arc length between nodes (for the arc-length window), and the route length of a given route (for the route-length window) cannot be decreased for the scheduling.

The fixed type window can be further classified as cumulative or non-cumulative windows. *Total route-length window* and *time window* are cumulative type because any change in an arc length between nodes or in the service time at a node affects the total length and the total time to reach another node, respectively. While the *arc-length window* is the non-cumulative type as an arc length is not dependent on the length of other arcs.

We assume that all the arcs ending at a specific node have the same window, hence the arc-length window is given to each customer.

The lower and upper bounds of the demand window of a node tell the minimum and maximum amount of commodity a vehicle carries after serving the node, respectively. A demand window has a lower limit of zero, and an upper limit of $D-d$, where D is the vehicle capacity and d is the demand quantity of the customer. After relaxation, the demand window becomes $[0,D-d+\delta]$, implying the customer is willing to get less than the originally planned quantity by δ. For the number of long tours window, the arc length can be defined as the travel distance, time, or the number of customers per route. Its upper limit is the number of long tours of the assigned vehicle in the same route of the customer. After relaxation, the window becomes $[0,b+\delta]$. The last two windows discussed here are adjustable type because the delivery quantity and the actual number of long tours can be decreased. These windows are explained in Table 1.

2.3. Problem formulation

There are K vehicles, each with capacity D, serving a set of n customers V'. The depot and V' form a new set of customers, V, and a network $G = (V,A)$, where A is the set of arcs connecting customers with each other and with the depot. Two nodes represent the depot: node 0 and $n+1$. A feasible vehicle route is a path in G that starts from node 0 and ends at node $n+1$ meeting all resource windows. There are L resource window types. Customer i has resource window, $\left[a_i^l, b_i^l\right]$ for resource l, where $a_i^l\left(b_i^l\right)$ is the lower (upper) bound of the resource window. Node i has resource consumption requirement s_i^l for resource l. The resources consumption requirement are the same for all vehicles.