Computers & Operations Research 39 (2012) 151–163

Contents lists available at ScienceDirect

Computers & Operations Research

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

An improved LNS algorithm for real-time vehicle routing problem with time windows

Lianxi Hong *

Computer Science and Engineering School, Jimei University, Xiamen 361021, Fujian, China

ARTICLE INFO

Available online 2 April 2011

Keywords:
Dynamic vehicle routing problem
Hard time window
Large neighborhood search
Heuristics

ABSTRACT

This paper studies the dynamic vehicle routing problem with hard time windows (DVRPTW). The main study course of this problem was briefly reviewed. The solving strategy and algorithm of the problem are put forward. First of all, DVRPTW problem is decomposed into a series of static VRPTW. When and how to decompose the DVRP is the issue, that must be addressed. An event-trigger mechanism has been proposed and used to decompose the DVRPTW into a series of system delay-snapshots. The trigger event to be adopted is a new request arrival during the stable operation. And each snapshot is regarded as a static VRPTW. Whether each static VRPTW can quickly and efficiently be solved within a given time or a shorter time, i.e. the solving time is another issue for the DVRPTW. In the solving process, how to merge the latest requirement to the current solution is the third issue that must be solved. An improved large neighborhood search (LNS) algorithm is proposed to solve the static problem. Utilizing the remove–reinsert process of the LNS, the latest request nodes are regarded as a part of the removed nodes; these nodes can be inserted into the original or current solution in good time in the reinsertion process; meanwhile, its computing speed is high and effective for it does not need to resolve primal problem each time. Computational results of a large number of test problems, which cited from Solomon′s static benchmarks and Lacker's dynamic data set, show that our method is superior to other methods in most instances.

1. Introduction

In the distribution and supply chain management and many other areas, the vehicle routing problem (VRP) has played a very important role. During the past five decades, many researchers have engaged to the research on the various types of vehicle routing problems and have made a great achievement. And the most of them have aimed at the static VRPs, all their information is assumed to be known and not to be changed in whole process. However, most of vehicle routing problems are dynamic in the real world. Dispatchers often need to readjust the vehicles′ routes to improve vehicles′ efficiency and enhance service quality when some accidents or unexpected incidents occur. With advances in modern communication technology to enable people to quickly access and process real-time data, the dynamic vehicle routing problem (DVRP) is being given more and more attention.

Relative to the static problem, the dynamic problem has many notable features [1]. They include that time dimension is essential, future information is imprecise or unknown, rerouting and reassignment decisions may be warranted, and faster solution speed is necessary and so on. In particular, it must be dynamically given the decision-making based on incomplete and uncertain and changing information. Thus, it is not possible for the decision maker to solve the entire problem at once [2]. Reviews on the problem can be found in Bertsimas and Simchi-Levi [3], Powell [4], Gendreau and Potvin [5], Psaraftis [1,6], and Ghiani et al. [7].

Usually, the dynamic vehicle routing problem is decomposed into a series of static VRP. Then the static solving algorithm is applied to each static VRP. A key issue is whether the algorithm gives a quick and effectively good feasible solution for each static one in a given time. There are two essential issues that must be solved. They are solving strategy and solving algorithm. Solution strategy must address how and when to decompose the DVRP. Two major classes of solving strategies have been found from the literatures. One is what we call short-sighted strategy and the other is forward-looking strategy. The forward-looking strategy attempts to predict future events in some areas according to the probability of events occurring in the solution process. However, it requires a large number of previous data which have the significant features. In such strategy, the stochastic nodes will

Abbreviations: DVRPTW, dynamic vehicle routing problem with hard time windows; VRPTW, vehicle routing problem with time windows; LNS, large neighborhood search; VNS, variable neighborhood search; C node, customer node; C_S node, static customer node; C_D node, dynamic customer node; M node, mobile node

* Tel.: +86 592 8726126; fax: +86 592 6181601.
E-mail address: lxhong@jmu.edu.cn

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.03.006

be regarded as known nodes into the current solution, in accordance with the probability of their occurrence. Usually, a stochastic node will be served if it raised its service request before the service vehicle arrived at it; it will be ignored otherwise. Some papers that used this strategy can be found in Powell [8], Slater [9], Godfrey and Powell [10,11], Bent and van Hentenryck [12,13], Larsen et al. [14], Mitrović-Minić et al. [15], etc.

In the short-sighted strategy, the solution process is based on known data and does not consider the unknown information and un-occurred events, even if some event will certainly happen. In the solution process, the node is merged into the current solution immediately once its request arrives. The papers that adopted the strategy can be found in Mitrović-Minić et al. [15], [16,17], Fleischmann et al. [18], Chen and Xu [19], Gendreau et al. [2], Li et al. [20], Larsen et al. [21], etc. Psaraftis [1] argued that the methodological base for solution techniques explicitly developed and designed for dynamic vehicle routing problem was scant, and that what could be a dynamic equivalent of the archetypal vehicle routing problem was not even defined, let alone formulated or solved at the time. In our view, the situation remains so.

There are many approaches for dealing with static problems in DVRPs. Gendreau and Potvin [5] focused mainly on problems motivated by courier service and demand responsive transportation systems, which typically evolve within a local service area over a relatively short period of time. The system is a typical real-time VRP system. Wan-rong Jih and Hsu [22] employed the hybrid genetic algorithm to solve the single-vehicle pickup and delivery problem with time windows and capacity constraints. Ichoua et al. [23] proposed an assignment strategy for the problem related to customer requests of the DVRP. Slater [9] presented an approach which is based upon demand forecasting and leads to the generation of phantom orders and phantom routes. Then the actual orders substitute for phantom orders in an on-line customer order process. The routing and scheduling method includes using both parallel tour-building and parallel insertion algorithms. Mitrović-Minić et al. [15] focused on the dynamic PDPTW of which future requests are not stochastically modeled or predicted. The solution process uses the rolling time horizon method was proposed by Psaraftis. Fleischmann et al. [18] considered a dynamic routing system that dispatches a fleet of vehicles according to customer orders arriving at random during the planning period and presented a planning framework for event-based dispatching and dynamic travel time information. Bent and van Hentenryck [12,13] adopted a multiple scenario approach (MSA) that can continuously generate routing plan for scenarios including known and future requests. Alvarenga et al. [24] employed the modified Hybrid Column Generation Heuristic in order to use the routes that have been generated during the optimization process to quickly generate new programs. Song et al. [25] presented a wasp-like agent strategy to decide when to deal with the real-time request and re-optimize the vehicle routes. Chen and Xu [19] studied a dynamic vehicle routing problem with hard time windows, in which a set of customer orders arrives randomly over time to be picked up within their time windows. The dispatcher does not have any deterministic or probabilistic information on the location and size of a customer order until it arrives. Then they proposed a column-generation-based dynamic approach to solve the problem. Gendreau et al. [2] proposed a neighborhood search heuristics to optimize the planned route of vehicles in a context where new requests, with a pick-up and a delivery location, occur in real-time. Ahnmmed et al. [26] and de Oliveira et al. [27] adopted the ant colony optimization system approach to solve dynamic vehicle routing problems with time windows. Djadane et al. [28] proposed a dynamic GA approach to solve the dynamic vehicle routing problems with flexible time windows and fuzzy travel times. Li et al. [20] introduced and studied the real-time vehicle rerouting problem with time windows, which is applicable to delivery and/or pickup services that undergo services disruption due to vehicle breakdowns. A Lagrangian relaxation based-heuristic that includes an insertion based algorithm is designed to obtain a feasible solution for primal problem while a dynamic programming based algorithm solves heuristically the shortest path problems with resource constraints that result from Lagrangian relaxation.

According to the current research situation, the problem that will be studied is described as follows. The problem is a network that contains a depot and N customer nodes (C node for short), where N is the maximum number of the nodes. The network is in the Euclidean plane, so the travel distance $d_{i,j}$ between any two nodes i and j is the straight-line distance between them, and the travel time $t_{i,j}$ between them is a constant, both them are known, symmetric and satisfy the triangle inequality: $d_{i,j} \le d_{i,l}+d_{l,j}$, $t_{i,j} \le \cap t_{i,l}+t_{l,j}$. Each C node denotes a service request. All requests will randomly arrive within a planning horizon. The depot has an associated deterministic time windows $[e_0,l_0]$ which corresponds to the planning horizon in the position $[x_0,y_0]$, and has k service vehicles $V=\{v_i, i=1,2,\ldots,k\}$. The maximum capacity of each vehicle is $\{vc_{v_i}, i=1,2,\ldots,k\}$. Every vehicle must depart from and return to the depot within $[e_0,l_0]$. For convenience, all vehicles are assigned a number starting from the $N+1$, i.e. $v_1=N+1, v_2=N+2, \cdots$. In the case of without causing confusion, the subscript of v_i will be ignored. Each C node has an associated location (x_i,y_i), a time window $[e_i,l_i]$, a service time st_i, a demand sd_i and a request arrival time ct_i. They are called the node parameters and can be expressed as a vector $c_i=(ct_i,sd_i,st_i,e_i,l_i,x_i,y_i)$, where $l_i \ge e_i, e_0 \le ct_i < l_0$ and $sd_i \le vc_n$, and $e_i,l_i,st_i,sd_i,x_i,y_i,ct_i,vc_n \in \mathbb{R}^+$, and n is the vehicle code that serves node i. Every node is exactly visited once by a vehicle during its time window is opening. We only consider the pickup or delivery service. The nodes can be divided into two classes according to their requesting time. One is the static nodes (C_S node for short) and the other is the dynamic nodes (C_D node for short). The nodes that have brought forward their requirements, before the first vehicle departs from the depot, are called the C_S nodes. The parameters of the C_S nodes are known before planning the routes and their $ct_i=0, i=1,2,\cdots$. The routes of the C_S nodes have been planned prior to the first vehicle leaving the depot. A node that puts forward its request after the first vehicle has left the depot and some vehicles are already on their way is called the C_D nodes. Within the planning horizon $[e_0,l_0]$, the C_D nodes will randomly call in for a request on-site pickup or delivery. The dispatcher does not have any advance knowledge of these nodes, either deterministic or probabilistic. And the parameters of a C_D node becomes to as a known when its request arrives. Following Lund et al. [29], the degree of dynamism of the system is defined as follows:

$$Dod = \frac{N_{dr}}{N_{dr}+N_{sr}} \times 100\% \tag{1}$$

where N_{dr} denotes the number of the C_D nodes and N_{sr} denotes the number of the C_S nodes. The object is to dispatch all vehicles over the planning horizon $[e_0,l_0]$ to cover all C_S nodes and all C_D nodes so as to minimize the total service cost which include total travel distance and the number of used vehicles under given Dod conditions.

We will adopt the short-sighted strategy to transform the DVRPTW into a series of static VRPTW. Static VRPTW has been widely discussed and studied in many literatures. Many heuristics have been proposed for this problem. They include the column-generation-based approach enhanced by strong valid inequalities proposed by Kohl et al. [30], Chen and Xu [19], and the branch-and-cut algorithm was proposed by Bard et al. [31], the Large Neighborhood Search (LNS) was proposed by Shaw [32] the Variable Neighborhood Search (VNS) was proposed by Hansen and Mladenovic [33], etc. Those algorithms can find optimal solutions to problems with up to 100 nodes. But most of them