Computers & Industrial Engineering 107 (2017) 211–222

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem

María Soto [a], Marc Sevaux [a,*], André Rossi [b], Andreas Reinholz [c]

[a] Université de Bretagne-Sud, Lab-STICC - UMR 6285 - CNRS, 2 rue de Saint Maudé, F-56321 Lorient, France
[b] Université d'Angers, LERIA - EA 2645 - CNRS, UFR Sciences, 2 Boulevard Lavoisier, 49045 Angers, France
[c] DLR, Institute of Solar Research, Linder Höhe, D-51147 Köln, Germany

ARTICLE INFO

Article history:
Received 26 November 2015
Received in revised form 15 March 2017
Accepted 17 March 2017
Available online 20 March 2017

Keywords:
CVRP
OVRP
MDOVRP
MNS
TS
Ejection chains
Path move

ABSTRACT

In this paper, we address the Multi-Depot Open Vehicle Routing Problem (MDOVRP), which is a generalization of the Capacitated Vehicle Routing Problem (CVRP) where vehicles start from different depots, visit customers, deliver goods and are not required to return to the depot at the end of their routes. The goal of this paper is twofold. First, we have developed a general Multiple Neighborhood Search hybridized with a Tabu Search (MNS-TS) strategy which is proved to be efficient and second, we have settled an unified view of ejection chains to be able to handle several neighborhoods in a simple manner. The neighborhoods in the proposed MNS-TS are generated from path moves and ejection chains. The numerical and statistical tests carried out over OVRP and MDOVRP problem instances from the literature show that MNS-TS outperforms the state-of-the-art methods.

1. Introduction

The Open Vehicle Routing Problem (OVRP) is a variant of the well known Capacitated Vehicle Routing Problem (CVRP) (Dantzig & Ramser, 1959), in which vehicles do not return to the depot. Each route ends at the last served customer. This problem was first introduce by Schrage (1981), then Sariklis and Powell (2000) introduced this problem as OVRP for the first time. One of the first applications of this problem is a special air cargo routing problem for FedEx introduced by Bodin, Golden, Assad, and Ball (1983). OVRP model is appropriate when the vehicles fleet is hired (Sariklis & Powell, 2000). For instance, in the transportation of handicapped persons (Soto, Sevaux, & Rossi, 2014), care centers for disabled people must provide daily transport to patients from their home to the care center as well as the return after treatment. These care centers work with specialized services companies in the transportation of handicapped persons, which have adapted vehicles for the transportation of disabled people. For other real-life applications of this kind of problem, readers are referred to Li, Golden, and Wasil (2007), Repoussis, Tarantilis, Bräysy, and Ioannou (2010, 2007), Tarantilis, Kiranoudis, and Vassiliadis (2004, 2005).

Furthermore, an excellent literature review about OVRP is provided in Zachariadis and Kiranoudis (2010) or in Li et al. (2007). Some exact methods have been proposed to tackle OVRP. For instance, Letchford, Lysgaard, and Eglese (2007) propose a Branch-and-Cut algorithm and Pessoa, de Aragão, and Uchoa (2008) propose a more complex Branch-and-Cut-and-Price strategy to solve the problem. Exact methods reach optimal solutions at the cost of prohibitively large computational times for real-life problem instances. Therefore, metaheuristic approaches seem a good alternative to produce high quality solutions in a reasonable computational time. Readers interested to know more about the metaheuristic dedicated to OVRP are referred to Reinholz and Schneider (2013), Sariklis and Powell (2000), MirHassani and Abolghasemi (2011), Li et al. (2007), Repoussis et al. (2010), Fleszar, Osman, and Hindi (2009), Zachariadis and Kiranoudis (2010), Li, Leung, and Tian (2012), Yu, Ding, and Zhu (2011), and Derigs and Reuter (2009).

There are many real-life transportation problems, where the vehicles can depart from several depots. In the logistic field, this problem is well-known as the Multi-Depot Vehicle Routing Problem MDVRP (Chao, Golden, & Wasil, 1993; Cordeau, Gendreau, & Laporte, 1997a; Lim & Wang, 2005; Lau, Chan, Tsui, & Pang, 2010; Renaud, Laporte, & Boctor, 1996; Wren & Holliday, 1972). Thus, the Multi-Depot Open Vehicle Routing Problem

* Corresponding author.
E-mail addresses: sotolima5@gmail.com (M. Soto), marc.sevaux@univ-ubs.fr (M. Sevaux), andre.rossi@univ-angers.fr (A. Rossi), andreas.reinholz@dlr.de (A. Reinholz).

http://dx.doi.org/10.1016/j.cie.2017.03.022
0360-8352/

MDOVRP is a variant of the MDVRP, which is more complex than the single-depot version (OVRP). The MDOVRP has been introduced by Tarantilis and Kiranoudis (2002) to tackle a distribution problem of fresh meat. They proposed a list-based threshold-accepting algorithm. Also, Liu, Jiang, and Geng (2014) present a mixed integer programming mathematical formulation and a hybrid genetic algorithm. Recently, Lalla-Ruiz, Expósito-Izquierdo, Taheripour, and Voss (2015) proposed a new mixed integer programming formulation for the MDOVRP, which improves the subtour elimination constraints proposed by Liu et al. (2014) and provides new sets of constraints.

In this paper, we propose a Multiple Variable Neighborhood Search hybridized with a Tabu Search called MNS-TS to tackle the MDOVRP. The structure of this paper is organized as follows. The next section presents the MDOVRP. Then, the following sections describe the ingredients of the MNS-TS. Section 3 introduces the Path Move, Section 4 explains the generation of our ejection chains and Section 5 presents the different neighborhoods we are using. Section 6 presents the Tabu Search dedicated to the MDOVRP. Our greedy algorithm to generate initial solutions is presented in Section 7. Section 8 summarizes our MNS-TS approach. Extensive computational results as well as the statistical tests are given in Section 9. Section 10 concludes the paper.

2. Multi-depot open VRP

This problem is defined on a complete directed graph $G = (V, A)$, where $M = \{1, \ldots, m\}$ is the set of depots and $N = \{1, \ldots, n\}$ is the set of customers. Thus, $V = M \cup N$ is the set of all vertices in the graph G, with $M \cap N = \emptyset$. The set of arcs is denoted by A. Each arc (i,j) in A represents connections between two vertices. $\forall i \in N, q_i$ is the demand of customer i. Depots have a zero demand, hence $q_i = 0, \forall i \in M$. Depots have enough goods to deliver to all the customers, and an unlimited fleet of homogeneous vehicles with a capacity Q. Each arc (i,j) is associated with the distance $d_{i,j}$ between the two vertices i and j. The traveling length of any vehicle route must be less than a given threshold D. The goal is to determine the vehicle routes which minimize the total traveling length satisfying the following constraints:

- each route starts at any depot and finishes at the last visited customer,
- each customer must be visited by exactly one vehicle,
- the total demand of the customers on the route of any vehicle cannot exceed its capacity Q,
- the total length of each vehicle route must not exceed D, the length limit.

A solution S of the MDOVRP is represented as a quintuplet $S = (\mathbb{X}, C, E, R, O)$. Where matrix $\mathbb{X}$ represents the vehicle routes, vector X_k is the vehicle route k and $x_{h,k} = i$ means that vertex i is the h-th vertex visited by vehicle route k. Vector C is the used capacity, $c_k \in \mathbb{R}^*_+$ is the capacity used by the vehicle route k. Vector E is the length of the vehicle routes, $e_k \in \mathbb{R}^*_+$ is the traveled distance performed by the vehicle route k. Vector R is the route assigned to customers, $r_i = k$ indicates that customer i is served by the vehicle k. Finally, vector O is the position of the customers in the routes, $o_i = h$ means that customer i is the h-th customer in the vehicle route r_i. Note that matrix $\mathbb{X}$, vectors R and O carry the same pieces of information, *i.e.* $x_{h,k} = i$ if and only if $r_i = k$ and $o_i = h$.

The objective function to minimize is the total traveling length and can be expressed as follows:

$$f(S) = \sum_{k} \sum_{h=1}^{|X_k|-1} d(x_{h,k}, x_{h+1,k}) \tag{1}$$

3. Path and path moves

This section introduces the *path move*, which is at the core of the ejection chains and the neighborhoods of MNS-TS. First, we define a *path* and a *path move*. Then, we presents the contribution length function, which computes the contribution length by moving a path. Finally, we define a function which allows us to determine when a path move is feasible or not.

3.1. Path

A *path* is a sequence of customers composed by one or more consecutive customers. A path is denoted by $P_i^\alpha, \forall i \in N$ and $\alpha \in \mathbb{N}^*$. It means that this path starts at customer i in the route r_i and visits α customers. A path only composed by one customer i is denoted by P_i.

For instance, in the following route:

$$5, 6, \underbrace{7}_{P_7}, 1, 11, \underbrace{3, 2, 10, 15}_{P_3^4}, 8, 9$$

we have path P_7 that is only composed by customer 7, and path P_3^4 that starts at customers 3 and is composed by four consecutive customers: 3, 2, 10 and 15.

We denote by $\ell(P_i^\alpha)$ the length of path P_i^α:

$$\ell(P_i^\alpha) = \sum_{h=o_i}^{o_i+\alpha-2} d(x_{h,r_i}, x_{h+1,r_i}) \tag{2}$$

Furthermore, we denote by $\ell_I(P_i^\alpha)$ the length of the reverse path, expressed by:

$$\ell_I(P_i^\alpha) = \sum_{h=o_i+\alpha-1}^{o_i+1} d(x_{h,r_i}, x_{h-1,r_i}) \tag{3}$$

Finally, Eq. (4) explicits $\delta(P_i^\alpha)$ which is the total demand of the customers that are part of P_i^α:

$$\delta(P_i^\alpha) = \sum_{h=o_i}^{o_i+\alpha-1} q_{x_{h,r_i}} \tag{4}$$

3.2. Path move

A *move* consists in removing a path P_i^α from its route to reinsert it after some customers j in the same route or not. There are two different ways of reinserting a path, because the vertex sequence of a path may be reversed. Reversion is represented by ω in $\Omega = \{1, 2\}$. Thus, ω is set to 2 if path P_i^α is reversed and inserted after customer j, otherwise ω is set to 1.

A *path move* is denoted by a triplet (P_i^α, j, ω), which means that path P_i^α is inserted in the route of customer j right after it, with the reversion or not indicated by ω.

With the goal of reducing the size of the neighborhoods presented in Section 5, for each customer i, we determine a neighborhood $\mathcal{N}(i)$ composed by the closest customers of i in G. For instance, the neighborhood of a vertex i can be composed by the vertices j such that the distance from vertex i to vertex j is less than a radius π, *i.e.*, $\mathcal{N}(i) = \{j \in V, d(i,j) \leqslant \pi\}$.

Furthermore, we define by $\mathcal{P}(S)$, the set of all possible path moves. We also define $\mathcal{P}_\alpha(S, L)$ as the set of path moves of size α that can be generated from a solution S and a list of customers L. Specifically, it is stated that $\mathcal{P}_\alpha(S, L) = \{(P_i^\alpha, j, \omega) \in \mathcal{P}(S), i \in L, j \in \mathcal{N}(i), \omega \in \Omega\}$. In the elementary situation defined by $\alpha = 1$ and $\omega = 1$, the set of possible path moves of size one is denoted by $\mathcal{P}(S, L)$.