European Journal of Operational Research 251 (2016) 44–52

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem☆

Hipólito Hernández-Pérez*, Inmaculada Rodríguez-Martín, Juan-José Salazar-González

DMEIO, Facultad de Ciencias, Universidad de La Laguna, La Laguna, Tenerife, 38271 Spain

ARTICLE INFO

Article history:
Received 20 November 2014
Accepted 26 October 2015
Available online 30 October 2015

Keywords:
Pickup-and-delivery
Hybrid approach
Traveling salesman
Local search

ABSTRACT

We address in this article the multi-commodity pickup-and-delivery traveling salesman problem, which is a routing problem for a capacitated vehicle that has to serve a set of customers that provide or require certain amounts of m different products. Each customer must be visited exactly once by the vehicle, and it is assumed that a unit of a product collected from a customer can be supplied to any other customer that requires that product. Each product is allowed to have several sources and several destinations. The objective is to minimize the total travel distance. We propose a hybrid three-stage heuristic approach that combines a procedure to generate initial solutions with several local search operators and shaking procedures, one of them based on solving an integer programming model. Extensive computational experiments on randomly generated instances with up to 400 locations and 5 products show the effectiveness of the approach.

1. Introduction

The *Multi-Commodity Pickup-and-Delivery Traveling Salesman Problem* (m-PDTSP) is a generalization of the *Traveling Salesman Problem* in which a capacitated vehicle based at a *depot* must visit a set of *customers*. Each location (depot and customers) must be visited exactly once by the vehicle. The travel cost between each pair of locations is known, and not necessarily symmetric. Each customer requires some given quantities of different products and/or provides some given quantities of other different products. A unit of a product collected from a customer can be supplied to any customer that requires this product. It is assumed that the vehicle must start and finish the route at the depot. Another visit to the depot is not allowed. The aim of the m-PDTSP is to find a Hamiltonian route for the vehicle such that it picks up and delivers all the quantities of the different products satisfying the vehicle-capacity limitation and minimizing the total travel cost. Since each customer is visited once, each unit of a product loaded on the vehicle stays on the vehicle until it is delivered. For that reason we say that the m-PDTSP is a *non-preemptive* problem.

The initial load of any product in the vehicle when leaving the depot is unknown, and must be determined within the optimization problem. The variant of the m-PDTSP where the initial load of any product is fixed can also be solved through the approach described in this paper with a slight modification of the instance. Note that, the initial load being unfixed, the vehicle is allowed to deliver a demand immediately when leaving the depot and collect the associated product afterwards. In other words, there are not a priori precedence relations between pickup and delivery locations of a commodity in the route. Still, the approach described in this paper can be adapted to the variant where the the initial load is required to be zero. This assumption is argued in Hernández-Pérez and Salazar-González (2014).

An application of the m-PDTSP occurs in the context of inventory repositioning, as pointed out by Anily and Bramel (1999). Assume that a set of retailers is geographically dispersed in a region. Often, due to the random nature of the demand, some retailers have an excess of inventory of some products while others need additional stock. In many cases the company may decide to transfer inventory from retailers with low sales to those with high sales. Determining the cheapest way to execute a given stock transfer (with the requirement that each location has to be visited exactly once) is the m-PDTSP.

Another application arises in the context of a self-service bike hiring system, where every night a capacitated vehicle must visit the bike stops in a city to collect or deliver bikes to restore the initial configuration of the system. Chemla, Meunier, and Wolfler-Calvo (2013) and Raviv, Tzur, and Forma (2013), among others, approached the case where the bikes are all identical (i.e., one product) as a 1-PDTSP. When there are different types of bikes (e.g., with and without baby chairs) the problem can be described as the m-PDTSP.

☆ This work was supported by the research project MTM2012-36163-C06-01, “Ministerio de Economía y Competitividad”, Spain .

* Corresponding author. Tel.: +34922845245; fax: +34922319202.
E-mail addresses: hhperez@ull.edu.es, hhperez@ull.es (H. Hernández-Pérez), irguez@ull.es (I. Rodríguez-Martín), jjsalaza@ull.es (J.-J. Salazar-González).

http://dx.doi.org/10.1016/j.ejor.2015.10.053
0377-2217/

The m-PDTSP is $\mathcal{NP}$-hard in the strong sense since it coincides with the TSP when the vehicle capacity is large enough. What is more, even checking the existence of a feasible solution for an instance with $m = 1$ is a strongly $\mathcal{NP}$-complete problem (see Hernández-Pérez & Salazar-González, 2004). The m-PDTSP was introduced by Hernández-Pérez and Salazar-González (2014). They proposed a Mixed Integer Linear Programming model for the m-PDTSP, and described a branch-and-cut algorithm able to solve instances with up to 30 customers and 3 commodities. Since exact algorithms are only able to cope with small instances, heuristic approaches are needed to tackle larger instances in practice. This is the main motivation of our paper.

A closely related problem to the m-PDTSP is the *Non-Preemptive Capacitated Swapping Problem* (NCSP), proposed by Erdoğan, Cordeau, and Laporte (2010). As in the m-PDTSP, the NCSP considers one depot, a set of customers, and several commodities with many sources and many destinations. In the NCSP, however, each customer concerns one commodity (either as pickup or delivery location) or two commodities (one as pickup location and another as delivery location). In addition to customer locations, the NCSP also includes transshipment locations, where some commodities can be temporarily dropped off. Customer and transshipment locations may be visited zero, one or two times by the vehicle, while the depot may be visited up to three times. The demand of a customer cannot be split and a commodity cannot be dropped off in an intermediate customer. The NCSP consists of finding a minimum-cost route satisfying all customer's requests. Erdoğan et al. (2010) describe a branch-and-cut algorithm to solve instances with up to 20 locations and 8 commodities, and 30 locations and 4 commodities. Bordenave, Gendreau, and Laporte (2009) present a branch-and-cut algorithm to solve the particular case of the NCSP where the vehicle capacity and customer demands are all equal to one. They solved instances with up to 200 locations and 8 commodities.

The *one-to-one m-PDTSP* is a particular case of the m-PDTSP where each commodity has one origin and one destination. It can be considered as a Dial-a-Ride routing problem without time window requirements. The one-to-one m-PDTSP assumes that the load of the vehicle when leaving the depot is zero, unless the depot is the source of a commodity. Hernández-Pérez and Salazar-González (2009) describe a branch-and-cut algorithm for this problem solving instances involving up to 24 customers and 15 commodities. Rodríguez-Martín and Salazar-González (2011) propose and compare several metaheuristic approaches to solve instances with up to 300 customers and 600 commodities.

Some articles dealing with one-commodity variants are the following. Chalasani and Motwani (1999) study the special case of the 1-PDTSP where the delivery and pickup demands are all equal to one. This problem is called *Q-delivery TSP*, where Q is the capacity of the vehicle. Anily and Bramel (1999) consider the same problem with the name *Capacitated TSP with Pickups and Deliveries*. Hernández-Pérez and Salazar-González (2007) present an exact algorithm for the 1-PDTSP solving instances with up to 200 customers. Hernández-Pérez, Rodríguez-Martín, and Salazar-González (2009) describe a hybrid algorithm that combines Greedy Randomized Adaptive Search Procedure and Variable Neighborhood Descent paradigms. Zhao, Li, Sun, and Mei (2009) propose a Genetic Algorithm that on average gives better results. Finally, Mladenović, Urošević, Hanafi, and Ilić (2012) describe a General Variable Neighborhood Search improving the best-known solution for all benchmark instances and solving instances with up to 1000 customers.

As in most of the articles dealing with TSP variants, we have decided to keep the assumption that each customer in the m-PDTSP must be visited once. However, the literature on vehicle routing includes articles that do not make this assumption. In particular, some authors address variants where a customer must be visited at most once (e.g. Fischetti, González, & Toth, 1997; Ghiani & Improta, 2000), or where a customer must be visited at least once (e.g. Archetti & Speranza, 2012; Nagy & Salhi, 2005; Nowak, Ergun, & White, 2009; Salazar-González & Santos-Hernández, 2015). There are also articles on related problems considering more than one vehicle, as in Psaraftis (2011), but, to our knowledge, in all of them each commodity must be transported from one source to one destination. Our paper is the first one proposing an approach for dealing with large instances of a capacitated pickup-and-delivery problem where each commodity may have several sources and several destinations.

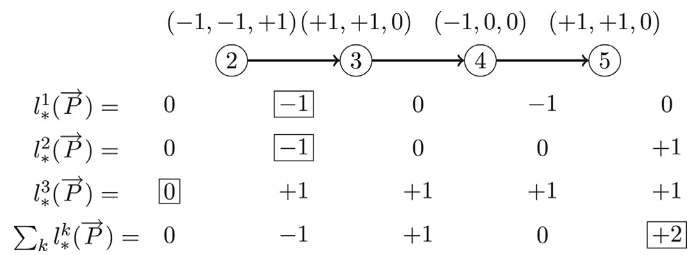

Fig. 1. Infeasible path when $m = 3$ and $Q = 3$.

The rest of the paper is organized as follows. Section 2 gives the formal definition of the problem and presents the notation used throughout the paper. Section 3 describes the heuristic algorithm to solve the m-PDTSP. Section 4 is devoted to the tuning of the algorithm's parameters. Computational results are shown in Section 5, and final remarks are made in Section 6.

2. Problem definition and notation

The m-PDTSP is defined on a complete directed graph $G = (V, A)$. The vertex set $V = \{1, \dots, n\}$ represents the locations. Vertex 1 is the depot and can be identified in the rest of the paper as a customer. For each pair of customers i and j we have the arc $a = (i, j) \in A$ and a travel cost c_{ij}. Let $K = \{1, \dots, m\}$ be the set of products. For each customer $i \in V$ and each product $k \in K$ let q_i^k be the demand of product k associated with i. When $q_i^k > 0$ customer i provides q_i^k units of product k and when $q_i^k < 0$ customer i requires $-q_i^k$ units of product k. We assume that $\sum_{i\in V} q_i^k = 0$ for all $k \in K$, i.e., each product is conserved through the route. The capacity of the vehicle is denoted by Q.

As mentioned above, contrary to what happens in the TSP, finding a feasible solution (optimal or not) for the m-PDTSP is a problem with a hard computational complexity. Nevertheless, checking if a given TSP solution is feasible for m-PDTSP can be done in $O(mn)$ time. Indeed, let us consider a path $\overrightarrow{P}$ defined by the vertex sequence $v_1, \dots, v_s$ for $s \le n$. Let $l_i^k(\overrightarrow{P}) := l_{i-1}^k(\overrightarrow{P}) + q_{v_i}^k$ be the load of the vehicle when coming out from v_i, considering that the vehicle enters customer v_1 with load $l_0^k(\overrightarrow{P})$. Notice that $l_i^k(\overrightarrow{P})$ could be negative if $l_0^k(\overrightarrow{P}) = 0$ and, therefore, the minimum quantity of load of commodity k for a feasible solution through the path $\overrightarrow{P}$ is $-\min_{i=0}^{s}\{l_i^k(\overrightarrow{P})\}$. With this notation, $\overrightarrow{P}$ is an infeasible path if

$$\max_{i=0}^{s}\left\{\sum_{k\in K} l_i^k(\overrightarrow{P})\right\} - \sum_{k\in K}\min_{i=0}^{s}\left\{l_i^k(\overrightarrow{P})\right\} > Q. \tag{1}$$

For example, consider an m-PDTSP instance with $m = Q = 3$ and customers 2, 3, 4 and 5 with demand vectors $(-1, -1, +1)$, $(+1, +1, 0)$, $(-1, 0, 0)$ and $(+1, +1, 0)$, respectively. The path defined by the customer sequence 2, 3, 4 and 5 is infeasible because

$$\max_{i=0}^{s}\left\{\sum_{k=1}^{m} l_i^k(\overrightarrow{P})\right\} - \sum_{k=1}^{m}\min_{i=0}^{s}\left\{l_i^k(\overrightarrow{P})\right\} = 4 > Q.$$

Fig. 1 illustrates the calculations done in this example.