Computers & Operations Research 51 (2014) 257–267

Contents lists available at ScienceDirect

Computers & Operations Research

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

An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows

Diego Cattaruzza [a,*], Nabil Absi [a], Dominique Feillet [a], Daniele Vigo [b]

[a] Ecole des Mines de Saint-Etienne, CMP Georges Charpak, CNRS UMR 6158 LIMOS, F-13541, Gardanne, France
[b] Alma Mater Università di Bologna, DEI - Dip. di Ingegneria dell'Energia Elettrica e dell'Informazione "Gugliemo Marconi", Viale Risorgimento, 2, 40136 Bologna, Italy

ARTICLE INFO

Available online 20 June 2014

Keywords:
Iterated local search
Multi-trip
Incompatible commodities
Split

ABSTRACT

The Multi-Commodity Multi-Trip Vehicle Routing Problem with Time Windows calls for the determination of a routing planning to serve a set of customers that require products belonging to *incompatible* commodities. Two commodities are incompatible if they cannot be transported together into the same vehicle. Vehicles are allowed to perform several trips during the working day. The objective is to minimize the number of used vehicles.

We propose an Iterated Local Search that outperforms the previous algorithm designed for the problem. Moreover, we conduct an analysis on the benefit that can be obtained introducing the multi-trip aspect at the fleet dimensioning level. Results on classical VRPTW instances show that, in some cases, the fleet can be halved.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The well known Vehicle Routing Problem (VRP) (Toth and Vigo [16] and Golden et al. [6]) is an $\mathcal{NP}-$hard combinatorial optimization problem where a set of geographically scattered customers has to be served by a fleet of vehicles. An implicit assumption of the VRP is that each vehicle can perform only one route in the planning horizon. Several practical situations allow vehicles to perform several trips during the working day. The problem that arises is the Multi-Trip VRP (MTVRP), (Cattaruzza et al. [4], Olivera and Viera [11], Mingozzi et al. [9]).

This paper studies a variant of the MTVRP, where different commodities need to be delivered to customers. Commodities are *incompatible*, i.e., they cannot be transported together into the same vehicle. On the other hand, the vehicles can transport different commodities in different trips.

The problem has been introduced by Battarra et al. [2] and called the Minimum Multiple Trip Vehicle Routing Problem (MMTVRP). The study was motivated by a real-world application, where a set of supermarkets requires deliveries of different commodities. To the best of our knowledge, no further study has been done on the MMTVRP.

Classical routing problems implicitly deal with different commodities. In particular, when vehicles can transport only one commodity, the problem is split in several sub-problems, one for each commodity, where a set of dedicated vehicles is available for the deliveries. On the other hand, if vehicles can carry different commodities at the same time, customer's demands of all commodities are collapsed into a single value, that represents its total request. Moreover, demands of different commodities are normalized into vehicle capacity units. As a consequence the resulting problem is a single-commodity VRP. Recently Archetti et al. [1] introduced a variant of the Split-Delivery VRP (SDVRP) where multiple customer visits are allowed only if the customer requires different commodities. In this case, commodities need to be considered explicitly.

In this paper we propose an effective Iterated Local Search (ILS) to solve the MMTVRP and a procedure *AdSplit* that turns permutations of customers into solutions. Moreover, a study is conducted on the benefit of introducing the multi-trip aspect in fleet dimensioning problems.

The paper is organized as follows. The problem and the notation are introduced in Section 2. Section 3 describes the ILS algorithm, while Section 4 presents the *AdSplit* procedure. Finally, Section 5 presents the results and Section 6 draws some conclusions.

* Corresponding author. Tel: +33(0)442616674; fax: +33(0)442616674.
E-mail addresses: cattaruzza@emse.fr (D. Cattaruzza), absi@emse.fr (N. Absi), feillet@emse.fr (D. Feillet), daniele.vigo@unibo.it (D. Vigo).

http://dx.doi.org/10.1016/j.cor.2014.06.006
0305-0548/© 2014 Elsevier Ltd. All rights reserved.

2. Problem definition and notation

The problem we consider has been introduced by Battarra et al. [2] and arises in the distribution of merchandise to supermarkets. The main characteristics of the problem are the presence of time windows (TW) and the fact that goods belong to different commodities which cannot be transported in the same vehicle at the same time. Overtime is not allowed. Moreover, the number of used vehicles is a variable that needs to be minimized. This problem has been called by Battarra et al. [2] the Minimum Multiple Trip Vehicle Routing Problem (MMTVRP).

More precisely, the MMTVRP can be defined on a complete undirected graph $G=(V,E)$, where $V=\{0,\dots,N\}$ is the set of vertices and $E=\{(i,j)|i,j\in V, i<j\}$ the set of edges. Vertex 0 represents the depot and vertices $1,\dots,N$ the customers. A set of commodities B has to be delivered to the set of customers. Commodities are *incompatible* with each other, that means they cannot be transported together in the same vehicle. Therefore, it can be supposed that each customer requires only one commodity: if a customer requires more than one commodity, it can be replicated as many times as the number of commodities he requires, associating with each replication one commodity and the corresponding quantity to deliver. Hence, each customer i requires a quantity Q_i of the commodity b_i to be delivered during a TW indicated by $[E_i, L_i]$. Vehicles are allowed to arrive at the location of each customer i before the corresponding time instant E_i and wait until E_i to start service. Service at customer takes S_i time units.

A homogeneous fleet of vehicles is located at the depot. Vehicles have a fixed capacity Q_b that depends on the commodity b. Moreover, the depot is open during the time interval $[E_{0b}, L_{0b}]$. The loading time S_{0b} at the depot is given by the sum of two terms. The first is a constant time S_0. The second is a trip-dependent value equal to a constant time S_b multiplied by the quantity of commodity b transported in the trip. Vehicles cannot operate longer than a given *spread time* ST, while the total duration of each trip must not exceed a spread time ST_b dependent on the transported commodity. Also routing costs depend on the transported commodity throughout cost C_b for each kilometer travelled. Finally, covering the distance D_{ij} that separates each pair of vertices i, j requires T_{ij} time units. Travelling times and distances do not coincide.

The set of trips assigned to the same vehicle is called *journey*. The MMTVRP calls for the determination of a set of trips and an assignment of each trip to a vehicle that minimizes the number of used vehicles (i.e., journeys) and, in case of ties, minimizes the routing cost. Moreover, it satisfies the following conditions:

(1) each trip starts and ends at the depot;
(2) a single commodity is delivered along each trip;
(3) each trip transporting commodity b starts not earlier than E_{0b} and ends not later than L_{0b};
(4) trips assigned to the same vehicle do not overlap in time;
(5) each customer is served exactly once;
(6) service at customer i must start in the range $[E_i, L_i]$;
(7) the sum of the demands of the customers in any trip delivering commodity b does not exceed Q_b;
(8) each trip takes less than ST_b when delivering commodity b;
(9) each journey takes no longer than ST.

To simplify notation, the letter ν will be used to indicate both the journey and the vehicle that performs it. When confusion can arise, details will be given. A solution will be indicated by the Greek letter ξ, while a trip by the letter σ. To indicate that journey ν is part of the solution ξ, we will use $\nu\in\xi$. Analogously, $\sigma\in\nu$ indicates that trip σ is assigned to journey ν. T_ν indicates the duration of a journey ν, while $T(\sigma)$ indicates the duration of trip σ. The duration of a trip (resp. journey) is defined as the difference between the moment the vehicle that performs it (resp. the last trip in it) is back at the depot and the moment loading operations for the trip (resp. first trip in the journey) start.

3. Algorithm

The only algorithm for the MMTVRP that we are aware of, has been proposed by Battarra et al. [2]. In this work a two-step heuristic is repeated in an iterated manner. The first step creates a set of trips generated by means of a heuristic for the VRP with time windows. The second step combines the obtained trips into feasible journeys and obtains a solution. The creation of trips makes use of a guidance mechanism. It penalizes the creation of numerous trips that overlap the same time interval and the creation of long-lasting trips. These trips do not facilitate the packing step.

In this section we present the iterated local search (ILS) procedure for the MMTVRP. The general scheme of any ILS is presented in Algorithm 1 (Lourenço et al. [8]).

Algorithm 1. General Iterated Local Search scheme.

1: Create a solution ξ_0
2: Apply LS to ξ_0 and obtain ξ^*
3: **while** Termination criteria are not met **do**
4: Perturb ξ^* to obtain ξ'
5: Apply LS to ξ' and obtain ξ'^*
6: **if** $\xi^{*'}$ is accepted **then**
7: $\xi^* = \xi^{*'}$
8: **end if**
9: **end while**

An initial solution ξ_0 is generated and improved by a local search (LS) procedure. The local optimum that is obtained is indicated by ξ^*. The following steps are repeated, until predetermined termination criteria are not met. The solution ξ^* is perturbed (modified) and a new current solution ξ' is obtained. The LS is applied to ξ' and a solution $\xi^{*'}$ is obtained. If $\xi^{*'}$ is accepted (for example, based on its quality) it becomes the new current local optimum ξ^*.

Perturbation plays a key role in the diversification of the search: small perturbations are likely to create a ξ' that falls back into ξ^* after LS is applied, resulting in an inefficient exploration of the search space. On the other hand, large perturbations make the ILS comparable to a multi-start algorithm.

Our ILS, indicated with $\mathcal{A}_{ILS}$, manages permutations Ψ of the N customers, usually called *giant tour* in the literature. Initially, a permutation Ψ_0 is created as explained in Section 3.2 and the number of vehicles M is set to a valid upper bound, for example to N.

The initial solution ξ_0 is obtained from the giant tour Ψ_0 by the *AdSplit* procedure (see Section 4). A first local optimum ξ^* is obtained by applying LS (Section 3.3). If ξ^* is not feasible, it undergoes the *Repair* procedure (Section 3.4). At each step, the current ξ^* is perturbed in order to obtain solution ξ'. A perturbation consists in crossing the permutation Ψ^* associated with ξ^*, with another permutation Ψ_1. The resulting permutation Ψ' undergoes *AdSplit* and ξ' is obtained. A new local minimum ξ'^* is obtained by applying LS and, if needed, the *Repair* procedure.

Download English Version:

https://daneshyari.com/en/article/475550

Download Persian Version:

https://daneshyari.com/article/475550

Daneshyari.com