Computers & Industrial Engineering 120 (2018) 146–159

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

# Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows

José Brandão

*Departamento de Gestão, Escola de Economia e Gestão, Universidade do Minho, Largo do Paço, 4704 – 553 Braga, Portugal*
*CEMAPRE, ISEG, University of Lisbon, Portugal*

ARTICLE INFO

*Keywords:*
Vehicle routing and scheduling
Iterated local search
Heuristics
Metaheuristics
Logistics

ABSTRACT

The problem studied in this paper is the open vehicle routing problem with time windows. This problem is different from the better known vehicle routing problem with time windows because in the former the vehicles do not return to the distribution depot after delivering the goods to the customers. For solving this problem an iterated local search algorithm was used, whose good results are mainly due to the kind of perturbations applied, in particular, ejection chains, and also to the use of elite solutions. The performance of this algorithm is tested using a large set of benchmark problems, containing 418 instances in total. The solutions obtained show that it is competitive with the best algorithms existing in the literature.

## 1. Introduction

The *open vehicle routing problem with time windows* (OVRPTW) consists in finding a set of minimum cost open routes, in order to serve a given number of dispersed customers, whose geographical location, demand, and time window for the delivery are known. Each route is travelled by one vehicle assigned to it, which starts the trip at the depot and visits each customer of the route according to a given schedule. After visiting the last customer of the route, the vehicle does not return to the depot, which is why the route is called 'open'.

In practice, we have an OVRPTW when a company uses hired vehicles for the distribution of its goods because, after visiting the last customer of the route, the hired vehicles end the trip and is no longer paid for by the company that hired them. On the other hand, when the vehicles are owned by the company, each vehicle ends the journey at the depot. The latter case is an example of the well-known VRPTW, which has been studied by many researchers and, consequently, dozens of papers have been published describing exact and approximation algorithms. As the discussion of the algorithms for the VRPTW is not part of the scope of this paper, we advise interested readers to consult the research of Bräysy and Gendreau (2005a, 2005b) and Baldacci, Mingozzi, and Roberti (2012). In addition, a recent survey of algorithms for different types of vehicle routing problems can be found in Braekers, Ramaekers, and Nieuwenhuyse (2016).

So far as we know, Schrage (1981) has been the first author to bring to attention the practical applications of the VRP with open routes. Later, Bodin, Golden, Assad, and Ball (1983) described a real problem of express airmail distribution in the USA, containing many practical features, such as delivery or pickup time windows, total route length, and capacity of the airplane. They have solved two routing problems separately, one for deliveries, and the other for the pickups, using the Clarke and Wright savings algorithm, which has been modified to take into account the open routes.

In the literature there are many other practical problems that can be formulated as OVRPTW, although, as expected, most of them contain other features which show the richness and the complexity of the real problems. As examples, we cite the paper of Russell, Chiang, and Zepeda (2008), which solves a problem of newspaper distribution, and the paper of Repoussis, Paraskevopoulos, Zobolas, Tarantilis, and Ioannou (2009a), which describes the solution of a lubricant distribution problem. Some other examples have been described in Fu, Eglese, and Li (2005) and Li, Golden, and Wasil (2007).

In spite of the many applications mentioned above and their economic importance, the research regarding specific algorithms for the OVRP, i.e., the OVRPTW without time windows constraints, only started about two decades ago. Since then, many papers have been published. Examples include: Sariklis and Powell (2000), Brandão (2004), Fu et al. (2005), Tarantilis, Kiranoudis, Ioannou, and Prastacos (2005), Letchford, Lysgaard, and Eglese (2007), Li et al. (2007), and Fleszar, Osman, and Hindi (2009), amongst others. Several other variants of the OVRP have been addressed by published papers as, for example, Yu, Jewpanya, and Redi (2016), who studied the OVRP with cross-docking and Soto, Sevaux, Rossi, and Reinholz (2017), who studied the multi-depot OVRP. The same concept has also been applied to the arc routing problem, such as, for example, the study of Fung, Liu, and Jiang (2013). On the other hand, the only published papers about

*E-mail address:* sbrandao@eeg.uminho.pt.

https://doi.org/10.1016/j.cie.2018.04.032
Received 3 January 2017; Received in revised form 13 April 2018; Accepted 18 April 2018
Available online 22 April 2018

the OVRPTW, without other side constraints, are those of Repoussis, Tarantilis, and Ioannou (2007) and Repoussis, Tarantilis, and Ioannou (2009b). Besides these two papers, that of Vidal, Crainic, Gendreau, and Prins (2014) studies several variants of the VRP, including the OVPRTW, as does the working paper of Kritzinger, Tricoire, Doerner, Hartl, and Stützle (2012).

The constraints taken into account in the OVRPTW are the following: *(i)* all the vehicles have the same capacity, which must not be exceeded, and each one is assigned to one route; *(ii)* a vehicle route starts at the depot and finishes at a customer; *(iii)* customers' demand must be satisfied; *(iv)* each customer must be visited only once; *(v)* the delivery starts within the time window, i.e., a time interval, imposed by the customer, and therefore the vehicle has to wait on arriving before the beginning of that interval; *(vi)* the total travelling time of each route, including waiting and unloading time, cannot exceed a driver's working day.

The objective is to minimise the number of vehicles, subject to the constraints just mentioned and, for a given number of vehicles, to minimise the total distance travelled by the vehicles. In practice, the travelling time is, in general, more expensive than the travelling distance on account of the driver's wage. In spite of this, we decided to use distance as the second objective. The reason for this is that other authors do so, and we wish to compare our results with theirs. Independently of using as second objective the distance or the time, most researchers agree that the first objective is far more important than the second one. This means that, in general, the fixed cost of one more vehicle and the wage of the corresponding driver exceed any possible savings in the distance travelled.

Another interesting line of research could be the use of multiple objective combinatorial optimization (MOCO) for solving the OVRPTW, considering a set of different objectives, without any relation of preference between them. In general, no single point exists that minimises all the objectives because they are conflicting. Therefore, a MOCO method has to generate the set of all non-dominated solutions, which are also called Pareto optimal, or, at least, a representative subset of them. A non-dominated solution is defined as a solution that is better than any other in relation to at least one objective, without being worse in relation to all the remaining ones. The purpose of the algorithm presented in this paper is not to find the set of Pareto optimal solutions for the OVRPTW with the two objectives defined above. Nevertheless, it can yield two non-dominated solutions (in fact, they are approximate, as the algorithm is not exact) by reversing the priorities of the two objectives defined above. This will be done for one set of test problems, in order to just show the influence on the solutions of this hierarchy of objectives.

The OVRPTW is a NP-*hard* combinatorial problem, as each route is a Hamiltonian path with time windows (HPTW), as has been proven, for example, by the research of Syslo, Deo, and Kowaklik (1983), where the HPTW is converted into a travelling salesman problem with time windows (TSPTW), which is a well-known NP-*hard* problem.

The OVRPTW is very difficult to solve, mainly due to the presence of time windows constraints. Savelsbergh (1988) proved that even the problem of determining whether there is a feasible tour for the TSPTW is NP-*hard*. This difficulty is observed when devising heuristics, as some concepts, such as closeness between customers, lose most of their meaning, especially if the time windows are tight.

The algorithm created by Repoussis, Tarantilis, and Ioannou (2007) is a sequential insertion heuristic. According to the authors, the performance of this heuristic is due to the time window based criteria that incorporates a look-ahead feature. These criteria are used for the selection of the next customer to enter the route under construction, and for choosing where it will be inserted in the route.

Repoussis, Tarantilis, and Ioannou (2009b) use an evolutionary algorithm where, at each generation, the population of offspring is produced exclusively through mutation. Next, each offspring is improved by a tabu search algorithm.

Vidal et al. (2014) use an algorithm called unified hybrid genetic search, which contains a set of problem-independent procedures, which is embedded in a solution framework that links the algorithm with the specific attributes of each variant of a vehicle routing problem. Therefore, this framework allows the solution of a large number of different types of vehicle routing problems, including the OVPRTW.

Kritzinger et al. (2012) also developed what they call a unified variable neighbourhood search, which is capable of solving several kinds of vehicle routing problems with a fixed fleet size.

In this paper, we introduce an iterated local search algorithm (ILSA) for solving the OVRPTW. As with any other metaheuristic, in order to find high quality solutions, the iterated local search requires a good compromise between the diversification and intensification of the search. The diversification is obtained by exploring regions of the solution space that are far apart. This is the case, for example, if two OVRPTW solutions, $s_1$ and $s_2$, have no arcs in common. The intensification results from exploring thoroughly the vicinity of a given solution. This happens, for example, if $s_2$ is obtained from $s_1$ by deleting one arc and adding a new one, i.e., by only changing a few arcs of $s_1$ at a time, better solutions than $s_1$ may be found in its close neighbourhood.

The main contribution of this paper is the proposal of a set of procedures that act together originating an effective balance between diversification and intensification, allowing to find high quality solutions in a short computing time. These main features and procedures are the following: ejection chain moves, elite solutions, a proximity concept that takes into account time and space, a neighbourhood move restriction that takes into account this proximity, and also three solution phases, each with a different objective function. On the programming side, the characteristics of the problem are taken into account in such a way that the computational execution of the algorithm is considerably efficient.

The remainder of this paper is organised as follows. Section 2 presents the mathematical model, and Section 3 describes the ILSA, starting with a short presentation of the methodology on which it is based. Next we describe the specific main features of the algorithm and in the end we provide the outline of the algorithm, in order to show how the different components work together. The computational experiments are described in Section 4, and the final conclusions are presented in Section 5.

## 2. Notation and mathematical formulation

The customers and the depot are represented by a set of vertices, $V = \{0, 1, \ldots, N\}$, where 0 is the depot, and $N$ is the total number of customers. The set of arcs that connect the $V$ vertices is called $A$.

$d_{ij}$: travel distance between $i$ and $j$, $(i, j)\ A$. We assume that the matrix $(d_{ij})$ is symmetric and satisfies the triangle inequality, i.e., $d_{ij} = d_{ji}$, and $d_{ij} \le d_{ik} + d_{jk}$, for all $i, j, k\ V$. However, as the vehicles are not allowed to return to the depot, we set $d_{i0} =\ $, $i \in V \setminus \{0\}$.
$u_i$: time of unloading at vertex $i$, $i\ V$, $u_0 = 0$.
$t_{ij}$: travel time between $i$ and $j$, plus $u_i$, $(i, j)\ A$. We assume that $d_{ij} = t_{ij} - u_i$, $(i, j)\ A$.
$e_i$: earliest starting time of the delivery at vertex $i$, $i\ V$.
$l_i$: latest starting time of the delivery at vertex $i$, $i\ V$.
$q_i$: demand of vertex $i$, $i\ V$, $q_0 = 0$.
$m$: number of available vehicles. The vehicles are identical, and the number of vehicles available is unlimited. The minimum number required is given by (2) and the maximum is $N$.
$F$: fixed cost of each vehicle. Since the first objective is to minimise the number of vehicles used, the value of $F$ in (3) should be a large constant (for example, greater than the sum of the distances between all the vertices).
$Q$: capacity of each vehicle.
$\tau$: maximum daily driving time of a driver. Since our test problems are taken from the VRPTW, we assume that $\tau = l_0 - e_0$.