Computers & Operations Research 62 (2015) 23–35

Contents lists available at ScienceDirect

Computers & Operations Research

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

A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading

Marilène Cherkesly [a,d,*], Guy Desaulniers [a,d], Gilbert Laporte [b,c,d]

[a] École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7
[b] HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
[c] CIRRELT, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7
[d] GERAD, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

ARTICLE INFO

Available online 11 April 2015

Keywords:
Vehicle routing with pickup and delivery
Last-in-first-out loading
Genetic algorithm
GRASP
Adaptive large neighborhood search

ABSTRACT

In this paper, we solve the pickup and delivery problem with time windows and last-in-first-out (LIFO) loading. LIFO loading minimizes handling while unloading items from the vehicle: the items are loaded according to a linear stack structure, and an item can only be delivered if it is the one on top of the stack. Three exact branch-price-and-cut algorithms are available for this problem. These can solve instances with up to 75 requests within one hour. We propose a population-based metaheuristic capable of handling larger instances much faster. First, a set of initial solutions is generated with a greedy randomized adaptive search procedure. For each of these solutions, local search is applied in order to first decrease the total number of vehicles and then the total traveled distance. Two different strategies are used to create offspring. The first selects vehicle routes from the solution pool. The second selects two parents to create an offspring with a crossover operator. For both strategies, local search is then performed on the child solution. Finally, the offspring is added to the population and the best survivors are kept. The population is managed so as to maintain good quality solutions with respect to total cost and population diversity. Computational results on medium to large instances confirm the effectiveness of the proposed metaheuristic.

1. Introduction

This paper proposes a population-based metaheuristic for the pickup and delivery problem with time windows and last-in-first-out (LIFO) loading (PDPTWL). In the pickup and delivery problem (PDP), a set of vehicles is used to complete several requests. A request corresponds to transporting goods (or items) from a pickup node to a delivery node. The LIFO policy means that when a pickup node is visited, its corresponding item is loaded on top of a linear stack, and an item can only be delivered if it is on top of the stack. Fig. 1 depicts two vehicle routes where 0^+ and 0^- represent the depot at the beginning and the end of the route, 1^+ and 1^- represent the pickup and the delivery nodes for item 1, and 2^+ and 2^- represent the pickup and delivery nodes for item 2. The first route respects the LIFO policy, but not the second one. In route 2, item 1 is delivered when item 2 is on top of the stack, meaning that the LIFO policy is not respected. Each item has a specified load, and each pickup or delivery node has a given service time and a time window during which service must start. We consider an unlimited fleet of identical capacitated vehicles. A vehicle route is feasible if it respects (i) the vehicle capacity, (ii) the time windows, and (iii) the LIFO policy. Note that there is a single depot, and each vehicle route starts and ends at the depot. Travel costs are proportional to the total traveled distance. The PDPTWL consists of first minimizing the number of vehicles used, and then the total distance traveled, subject to the feasibility constraints.

To the best of our knowledge, the PDPTWL has only been studied by Cherkesly et al. [10] who have developed three exact branch-price-and-cut algorithms which can solve instances with up to 75 requests within one hour of computation time. In addition, a number of heuristics have been proposed for variants of the problem, namely the vehicle routing problem with time windows (VRPTW) (see [5,6,29] for a survey, and [30] for a state-of-the-art heuristic), the pickup and delivery problem with time windows (PDPTW) (see [4,19,20] for a survey, and [3,16,25] for recent heuristics), the traveling salesman problem with pickup and delivery and LIFO loading (see [7,8,17] for recent heuristics), the pickup and delivery problem with LIFO loading (see [1,9,15,17] for recent heuristics), and the traveling salesman problem with pickups, deliveries and handling costs (see [2] for a branch-and-cut algorithm and [12] for a heuristic).

* Corresponding author at: GERAD - HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7.
E-mail address: marilene.cherkesly@polymtl.ca (M. Cherkesly).

http://dx.doi.org/10.1016/j.cor.2015.04.002
0305-0548/

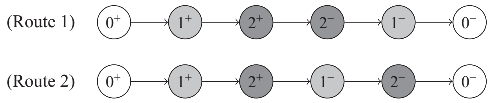

Fig. 1. The LIFO policy is respected in route 1, but not in route 2 because item 1 cannot be delivered without first removing item 2 from the vehicle.

Among the algorithms put forward for the PDPTWL variants, two main heuristic search principles emerge and will constitute the basis of this study. The first is the population-based heuristic of Vidal et al. [30] which can solve many variants of the VRPTW, namely the periodic VRPTW, the multi-depot VRPTW, and the site-dependent VRPTW. One important feature of this algorithm is the population management strategy which allows to diversify the solution pool. The second is the adaptive large neighborhood search (ALNS) of Ropke and Pisinger [25] for the PDPTW. The ALNS performs a local search by first removing requests and then reinserting them. The algorithm chooses at each iteration one of three removal operators and one of two reinsertion operators, according to their past performance.

The main objective of this paper is to propose a population-based metaheuristic capable of solving large-sized instances of the PDPTWL. In this algorithm, a set of initial solutions is obtained through the application of a greedy randomized adaptive search procedure (GRASP). Two population-based methods are then used to generate offspring. The first method combines routes from the solution pool, whereas the second applies an adapted order crossover operator. In the second method, a diversification strategy inspired by that of Vidal et al. [28] is used to update the solution pool. Local search based on the ALNS principle is then performed on each solution in order to first minimize the number of vehicles, and then the total traveled distance. Computational results are reported for instances with 30–300 requests, and show that the second method used to generate offspring produces better quality solutions.

The remainder of this paper is structured as follows. Section 2 presents the mathematical notation used. Section 3 describes the construction of initial solutions, the population-based metaheuristic, and the local search operators. Computational results are reported in Section 4, and conclusions follow in Section 5.

2. Problem description

Let $I=\{1,\dots,n\}$ denote a set of n items, also called requests, and let $P=\{1^+,\dots,n^+\}$ and $D=\{1^-,\dots,n^-\}$ represent the sets of pickup and delivery nodes. With each request $i\in I$ are associated a pickup node $i^+\in P$ and a delivery node $i^-\in D$. The depot is represented by two nodes 0^+ and 0^- which are respectively called the origin and the destination depot. The PDPTWL can be defined on a directed graph $G=(N,A)$, where $N=P\cup D\cup\{0^+,0^-\}$ is the set of nodes and A is the set of arcs. An arc $(i,j)\in A$ must respect the precedence constraints, and the LIFO policy, i.e., for each request $i\in I$, the arc (i^-,i^+) is not generated, and for each pair of requests $i,j\in I$ where $i\neq j$, the arc (i^+,j^-) is not created. Note that because the algorithm allows intermediate solutions containing infeasible routes with respect to time windows or capacity constraints, we allow arcs that violate the time windows or the capacity constraints.

For each request $i\in I$, q_i represents the load of the items to be picked up at node $i^+\in P$ and delivered at node $i^-\in D$. We denote by $q_{i^+}>0$ the load picked up at node $i^+\in P$, and by $q_{i^-}=-q_{i^+}$ the load delivered at node $i^-\in D$, with $q_i=0$ if $i\in\{0^+,0^-\}$. A time window $[a_i,b_i]$ is associated with each node $i\in N$, where a_i and b_i represent the earliest and the latest times at which the service can begin at node i, and waiting before the beginning of the time window is allowed. The time windows of the origin and the destination nodes 0^+ and 0^- are unconstraining. An unrestricted set of K identical vehicles of capacity Q is available. With each arc $(i,j)\in A$ are associated a nonnegative travel distance d_{ij}, and a nonnegative travel time t_{ij} which includes the service time at node i if any. We assume that the triangle inequality holds for travel distances and travel times.

Let $\mathcal{R}$ be the set of routes in a solution, and let $R\in\mathcal{R}$ be a route that can be denoted as $R=(i_0=0^+,i_1,i_2,\dots,i_m=0^-)$, where i_ρ is the ρth node visited in R. For each visited node $i_\rho\in R$, $1\le\rho\le m$, we compute the total load of the vehicle after visiting it as $l(i_\rho)=l(i_{\rho-1})+q_{i_\rho}$, with $l(i_0)=0$. We define $t(i_\rho)=\max\{t(i_{\rho-1})+t_{i_{\rho-1},i_\rho},a_{i_\rho}\}$ as the time at which service starts at node i_ρ, $1\le\rho\le m$ with $t(i_0)=0$. Note that because we accept infeasible intermediate solutions, it is possible that $t(i_\rho)>b_{i_\rho}$, i.e., the service at node i_ρ could start after the end of its time window. Thus, we define $\tau(i_\rho)=\min\left\{\max\{\tau(i_{\rho-1})+t_{i_{\rho-1},i_\rho},a_{i_\rho}\},b_{i_\rho}\right\}$ the service start time with time-warp at node i_ρ, $1\le\rho\le m$, with $\tau(i_0)=0$, and $w(i_\rho)=\max\{0,\tau(i_{\rho-1})+t_{i_{\rho-1},i_\rho}-b_{i_\rho}\}$ the time-warp needed at this node to respect the time windows, with $\tau(i_0)=0$. This concept was introduced by Nagata et al. [18] and extended by Vidal et al. [30] to be applied to all operators used in local search algorithms for routing problems, and is used when a vehicle arrives after the time window of a customer in order to reach the end of the time window. Each route R is associated with a set $I_R=\{i\in I\,|\,i\in R\}$ of completed requests, where $i\in R$ indicates that request i is served in route R. We denote by $d(R)=\sum_{\rho=0}^{m-1}d_{i_\rho,i_{\rho+1}}$ the total distance of route R, by $q(R)=\sum_{\rho=0}^{m}\max\{0,l(i_\rho)-Q\}$ the excess capacity of route R, and by $w(R)=\sum_{\rho=0}^{m}w(i_\rho)$ the time window violation of route R. If a route is feasible with respect to capacity, then $q(R)=0$; otherwise $q(R)>0$. Similarly if a route is feasible with respect to time windows, then $w(R)=0$; otherwise $w(R)>0$. Each route R has a cost

$$c(R)=d(R)+\alpha q(R)+\beta w(R), \tag{1}$$

where α and β are positive user-defined parameters.

Let $\mathcal{S}_f$ denote the set of feasible solutions, $\mathcal{S}_i$ the set of infeasible solutions, and $\mathcal{S}=\mathcal{S}_f\cup\mathcal{S}_i$ the solution pool. For each solution $S\in\mathcal{S}$, we denote by $\mathcal{R}_S$ the set of all routes in S. Each solution S has a cost $c(S)=\kappa|\mathcal{R}_S|+\sum_{R\in\mathcal{R}_S}c(R)$, where κ is a positive user-defined parameter, and $|\mathcal{R}_S|$ is the number of vehicles used.

The PDPTWL consists of determining a set of feasible routes covering exactly once each request with respect to capacity constraints, time windows, and the LIFO policy such that the number of vehicles is first minimized, and then the total traveled distance is minimized.

3. Description of the metaheuristic

We now describe the population-based metaheuristic we have designed for the PDPTWL. It proceeds in three phases. The first phase consists of creating an initial solution pool by means of a GRASP, a concept introduced by Feo and Resende [13,14]. The cost evaluation for each request is based on a savings criterion. Each solution goes through a local search phase to first minimize the number of vehicles, and then the total traveled distance. The GRASP generates feasible solutions only, but infeasibility is allowed in the local search phase. The second phase consists of creating additional solutions by selecting routes from different solutions and creating an offspring. Local search is applied to the offspring. Finally, the third phase consists of selecting two parents, and creating offspring by means of an adapted crossover operator. Each offspring is educated through local search. Note that the first phase of the algorithm is essential because it generates the initial solution pool, but the second and third phases are not. Thus, several variants of the algorithm are possible. For example one