Adaptive Large Neighbourhood Search for a Pickup and Delivery problem with time windows*

Matthieu Fagot[1,2], Laure Brisoux Devendeville[1], Corinne Lucet[1]

[1] Laboratoire MIS (UR 4290) Université de Picardie Jules Verne, France
1 Chem. du Thil, 80000 Amiens
{m.fagot, laure.devendeville, corinne.lucet}@u-picardie.fr
[2] Smile Pickup, France
5 rue des Indes noires, 80440 Boves
mfagot@smilepickup.com

* This project is supported by the PICKOPT project
(CIFRE 2021/0599 between Smile Pickup and the Laboratoire MIS).

1 Introduction

Smile Pickup manages a network of pickup points for large parcels as well as the planning of parcel transportation between stores and pickup points. The transport planning of Smile Pickup is similar to the NP-hard Pickup and Delivery Problems with Time Windows (PDPTW) to which specific constraints are added to match the company's activity constraints. Smile Pickup problem (SPP) spans over a period of H consecutive days during which parcels are exchanged between pickup points and stores using a heterogeneous fleet of vehicle. Constraints taken into account are multiple strict time windows, capacity constraints, single tour per vehicle and penalty for storing parcel before delivery. The objective is to minimise vehicle usage cost, total distance travelled, number of undelivered parcels and storage penalty. For the company, these objectives make it possible to reconcile economic profitability and quality of service. The specificity of the problem lies in the fact that the routes can pass through the same points multiple times and that each point can be both the departure and the destination of several parcels. To our knowledge, there is currently no publication directly addressing this last constraint without resorting to the duplication of the points and thus reducing it to a classic PDPTW problem. This solution is not very viable in practice given the high number of parcels we aim processed. In this article, we will describe the Smile Pickup problem (section 2) followed by a brief presentation of the Adaptative Large Neighbourhood Search algorithm developed (section 3) as well as an ε-greedy movement selection procedure 4. We will then conclude by presenting our results (section 5) and future works (section 6).

2 Smile Pickup Problem

The problem faced by Smile Pickup spans over a total of $H \in \mathbb{N}$ consecutive days during which parcels need to be transported between places using the fleet of vehicles.

Places. The set of places is divided in three subsets: depots where vehicles start and end their day, stores and pickup points which exchange parcels. Each place has a set of strict time windows for each day. Outside those time windows pickups and deliveries are forbidden. Sets of time windows can be empty if a point is not open on a certain day. For the depots, the time windows model the opening hours during which vehicles may start and end their tours. Finally, we associate a travel distance and a travel duration to each pair of points. The first one are used to evaluate solutions while the second are used to verify time constraints.

Parcels. A parcel is made available in a store or a pickup point on a certain day and needs to be delivered to it's destination. The place of origin can hold the parcel for any duration required, but if the parcel remains undelivered beyond the expected time, additional fees for storage will be charged for each additional day. Solutions do not need to deliver all parcels but each undelivered parcel will incur additional charges.

Vehicles. For each day and each depot, a set of vehicles is available. Vehicles are given a usage cost and a capacity corresponding to the length of the trailer. The sum of the length of the parcels in a vehicle must never exceed this capacity. A vehicle can arrive early at a place even though it will have to wait until a time window opens before starting loading and unloading.

3 Adaptative Large Neighbourhood Search

The Adaptive Large Neighbourhood Search (ALNS) algorithm [3] is an improvement of the Large Neighbourhood Search (LNS) [2] algorithm in which phases of solution deterioration and improvement are alternated to explore the solution space. During these phases, movements are chosen according to a probability distribution and then applied to the solution to explore its neighbourhood. These phases are grouped into segments at the end of which the probability distribution is updated. The ALNS differs from the LNS by allowing the automatic adaptation of the probability distribution during execution. This technique is similar to reinforcement learning. Each iteration of ALNS allows the algorithm to move from a solution to a neighbouring one.

Each iteration is subdivided into four phases. The first phase consists of deteriorating the current solution using three movements: removal of a parcel, removal of a point, or removal of a route. Then the second phase improves the degraded solution by applying parcel insertion movements until the new solution becomes saturated. In the next phase, if the new solution is better or if it is accepted by the aspiration criterion then it becomes the new current solution. A Simulated Annealing (SA) method is used as the aspiration criterion. It accepts certain deteriorating solutions to escape local optima. The probability of accepting a new solution depends on the current solution and decreases throughout the execution. This third phase also hands out rewards to movements that improve the solution. The last phase aims to update the probabilities W_m^s associated with each movement m at the end of each segment s: $W_m^{s+1} = \rho W_m^s + (1-\rho)\frac{\pi_m}{\theta_m}$ with $\rho \in]0, 1[$. π_m is the sum of the rewards of the movement m obtained during the last segment while θ_m is the number of times it was used.

4 ε-greedy movement selection

To improve our movement selection throughout the algorithm execution, an ε-greedy strategy is used in the movement selection phase. ε-greedy is a simple algorithm that was first introduced in the context of reinforcement learning by Sutton and Barto in 1998 [4]. ε-greedy aims to balance exploration and exploitation. The selection procedure involves using ε-greedy to switch between selecting the best movement based on the weights and exploring other potential movements. Two exploration scenarios were compared, one using a uniform distribution for all movements (ALNS ε-greedy uniform), and the other selecting movements according to their respective weights (ALNS ε-greedy weighted).

5 Results

In this section, we present the results obtained on two different databases : the first one called PickOpt was obtained using a random instance generator. 135 instances were used containing between 30 and 1000 parcels distributed among up to 50 locations. The second one is the well known Li & Lim's benchmark [1]. These instances were adapted to suit the requirements of SPP, even though they were originally designed for PDPTW which is a subset of SPP. The 56 instances with 50 parcels where used in our tests.

Table 1: Comparison of our algorithms on both PickOpt and Li & Lim's benchmark.

	PickOpt	Li & Lim	Both
LNS without SA	29.33 / 30.37	0 / 0	20.73 / 21.47
LNS	28.44 / 41.48	0 / 0 .71	20.31 / 29.32
ALNS	33.26 / 30.37	26.07 / 30.36	31.15 / 30.37
ALNS ε-greedy uniform	36.52 / 36.30	32.50 / 50.00	35.34 / 40.31
ALNS ε-greedy weighted	36.74 / 36.30	30.71 / 44.64	34.97 / 38.74

Each experiment was run 10 times on an Intel(R) Xeon(R) CPU E5-2680 v4 with a maximum execution time of 30 minutes. Results are shown in Table. 1. The values indicate the percentile of the best solutions and best averages achieved by the algorithm on the benchmark. Observing the data presented in Table. 1, it appears that ALNS with ε-greedy uniform exploration is the best on both benchmarks, albeit with a small margin on ALNS with ε-greedy weighted. Interestingly, LNS produces a higher percentile for PickOpt in terms of the best average. This outcome is somewhat surprising, as there is an 8% difference in the percentile of the best solutions. However, this can be attributed to the fact that PickOpt instances are not sufficiently difficult to test the capabilities of our algorithms.

6 Perspectives

In this short article, we presented an ALNS algorithm on a real life problem encountered by Smile Pickup. We introduced the use of an ε-greedy algorithm for the movement selection procedure. In future works, we will diversify our movement pool to further investigate the impact of the reinforcement learning algorithm. Additionally, we intend to refine and complement our problem instances to better test the capacities of our algorithm. Finally, we would like to go further on the movement selection and adaptiveness by trying to learn how to better orchestrate the sequence of movements used to explore solution space.

References

[1] Haibing Li and Andrew Lim (2003): " A Metaheuristic for the Pickup and Delivery Problem with Time Windows." In: *International Journal on Artificial Intelligence Tools*, 12(02):173–186.

[2] David Pisinger and Stefan Ropke (2007): " Large Neighborhood Search." In: *Discrete optimization*, 4(1):73–87.

[3] Stefan Ropke and David Pisinger (2006): " An Adaptive Large Neighborhood Search heuristic for the Pickup and Delivery Problem with Time Windows." In: *Transp. Sci.*, 40(4):455–472.

[4] Richard S Sutton and Andrew G Barto. (1998): " Reinforcement Learning: An Introduction." In: *MIT press*.