Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints

Arnaud Malapert[2,3], Christelle Guéret[1], Narendra Jussien[2], André Langevin[3], and Louis-Martin Rousseau[3]

[1] École des Mines de Nantes, YRCCyN UMR CNRS 6597
[2] École des Mines de Nantes, LINA UMR CNRS 6241
[3] École Polytechnique de Montréal, CIRRELT
{arnaud.malapert,christelle.gueret,narendra.jussien}@emn.fr
{louis-martin.rousseau,andre.langevin}@polymtl.ca

Abstract. In this paper, a special case of the vehicle routing problem in which the demands consist in a set of rectangular two-dimensional weighted items is considered. The vehicles have a two-dimensional loading surface and a maximum weight capacity. These problems have a routing and a packing component. A framework to handle the loading of a vehicle is proposed. A Constraint Programming loading model based on a scheduling approach is developed. It is also shown that the non-overlapping rectangle constraint can be extended to handle a practical constraint called sequential loading constraint .

Key words: vehicle routing problem with loading constraints, packing problem, constraint programming.

1 Problem description

The *Pickup and Delivery Problem with Two-dimensional Loading Constraints* (2L-PDP) is defined[1] by a quintuplet $\mathcal{P} = (G, K, I, M, O)$ where G is an undirected graph, K represents the fleet of vehicles, I the set of customers, M the set of items and O the objective.
Each vehicle $K = \{1, \dots, m\}$ has a weight capacity P_k and a rectangular loading surface of width W_k and height H_k. The vehicle has a single opening for loading and unloading items. The demand of each client $i \in I = \{1, \dots, n\}$ consists in a set of m_i items to be carried from a pickup site i^+ to a delivery site i^-. $\mathcal{G}$ is a complete graph $\mathcal{G} = (V_0, E)$ where $V_0 = \cup_{i \in I}\{i^+, i^-\} \cup \{D\}$ and D is the depot. E is the set of edges (i, j) between any pair of vertices, with associated cost c_{ij}. Each item $m_i^j \in M$ is associated to a client i and has a unique index $j \in \{1, \dots, m_i\}$, a width w_i^j, a height h_i^j and a weight p_i^j. We assume that the set of items demanded by each client can be loaded into a single vehicle.
Split deliveries are not allowed, *i.e.* the entire demand must be loaded on the same vehicle.The aim is to find a partition of the clients into routes of minimal

[1] with the formalism introduced by Gendreau and *al.* [1]

total cost, such that for each route there exists a *feasible loading* of the items into the vehicle loading surface. Of course, this feasible loading must satisfy the capacity constraint.
A feasible loading is defined as follows: first, it must satisfy the packing constraint, items cannot overlap each other and they must fit completely into the vehicle; second, only *orthogonal* packing are allowed, each item has its edges parallel to the sides of the loading surface. Most works on 2L-CVRP (see section 2) consider *oriented* packing, *i.e.*, items cannot be rotated. In comparison to the traditional packing problem, the load must satisfy a practical constraint called *sequential loading* [2]. This constraint requests that, when visiting a client, his items can be unloaded from the vehicle by means of forklift trucks without having to move items belonging to other clients along the route. When this constraint is not required, allowing thus for re-arrangements of the items in the vehicle at the clients' sites, the problem has an *unrestricted loading*.

2 Related works

2.1 Vehicle routing problem with loading constraints

Most of the *Vehicle Routing Problem* (VRP) solving techniques are focused on the route construction and improvement. The capacity constraint is the only constraint related to the load of the vehicle. But many real-world transportation problems have to deal with both routing and packing.
The first transportation problem with loading constraints is the *Pickup and Delivery Traveling Salesman with LIFO Loading* (TSPLL) [3] where a single rear-loaded vehicle must serve a set of customer requests. Consequently, loading and unloading operations must be performed according to a Last-In First-Out (LIFO) policy.
The *Two dimensional Capacited Vehicle Routing Problem with Loading Constraints* (2L-CVRP) was addressed by Iori and *al.* [2], through branch-and-cut and later by [1, 4], through Tabu Search (TS) and Ant Colony Optimization (ACO). Gendreau and *al.* [1] adapt their formulation to solve *Three-dimensional Capacited Vehicle Routing Problem with Loading Constraints* (3L-CVRP) [5].
Moura and *al.* [6] propose also a framework to integrate *Vehicle Routing Problem with Time Windows* (VRPTW) and the *Container Loading Problem* (CLP). They propose two different methods: a sequential method and a hierarchical method. The sequential method relaxes the sequential loading constraint and allows split deliveries. Therefore, routing and loading are planned at the same time. The hierarchical method addresses the original problem and uses the CLP as a sub-problem of the VRPTW. These two methods use a greedy constructive heuristics for CLP. This heuristics allows to consider extra constraints such as cargo stability.

2.2 Two-dimensional Loading Problem

The *Two-dimensonal Loading Problem* (2LP) consists in finding a feasible loading for a given route. It can be seen as a *containment problem* with additional

constraints. The containment problem consists in packing a set of items in a single bin of given width and height. Three models are available to encode a solution of a packing problem: a permutation model, a coordinates model and an interval graph model.
Iori and *al.* [4] propose two approaches to solve the loading sub-problem for a given route: local search and truncated branch-and-bound. Both methods use a lower bound. If the lower bound can not prove infeasibility, they try to compute an upper bound. Local search methods are based on two permutation heuristics: *Bottom-Left* and *Touching Perimeter*. For each packing position, the heuristics checks the sequential loading constraint on already packed items. A failure can only be detected when there are no more feasible positions for an item or if the height of the packing is greater than the height of the loading surface.
At each node of the search tree, the truncated branch-and-bound tries to place each remaining item in a limited subset of positions. Backtracking is performed when an item cannot enter any corner point. The procedure is halted when it reaches a maximum number of backtracks, or when a feasible solution is found.

There is no CP methods to solve the loading problem, but some packing methods have been proposed. [7, 8] use the coordinates model whereas Moffit and *al.* [9] uses a meta-CSP based on the graph model. Our model integrates some of their techniques.

3 Loading model

The loading sub-problem is solved very often during search and some instances are infeasible. Our goal is to detect infeasible loading as quickly as possible, to escape from infeasible regions of the search space.
Our model uses lower bounds, redundant constraints issued from the scheduling field (see Section 3.1) and the sequential loading constraint (see Section 3.3).
Scheduling constraints use the notion of *tasks* and *resources* (see section 3.1).
Our model uses a pair of tasks to represent the origin and the dimensions of an item, it implies that we use the coordinates model. This representation allows

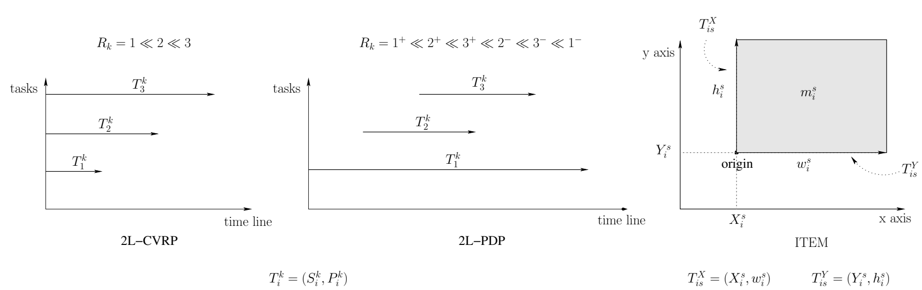

Fig. 1. The arrival time of vehicle k at the pickup (resp. delivery) site of client i is the starting (resp. ending) time of T_i^k. The starting time of T_{is}^X (resp. T_{is}^Y) is X_i^s (resp. Y_i^s) and its duration w_i^s (resp. h_i^s), this pair of tasks represents the packing of m_i^s.

an efficient interaction between packing and scheduling constraints.
The routes are also represented by a set of tasks which allows to add scheduling constraints (precedence, time windows,...) on or between clients. A routing solving technique must propagate its decisions on these tasks. These tasks are an interface to determine constraints on the loading due to the route.

Let i be a client, $T_i^k = (S_i^k, P_i^k)$ denotes a task of starting date S_i^k and processing time P_i^k associated to route R_k. If the route R_k does not serve the client i then $P_i^k = 0$, otherwise T_i^k must satisfy the properties presented in Figure 1. For each item, let defined $T_{is}^X = (X_i^s, w_i^s)$ (resp. $T_{is}^Y = (Y_i^s, h_i^s)$) a task associated to the dimension of m_i^s along X (resp. Y) axis (see Figure 1). Let $\ll$ denote a precedence relation between two nodes of the same route.

3.1 Scheduling constraints

In this section, we present the scheduling notions and constraints used in our model. A resource has a fixed capacity. A task has a requirement for each resource. The *cumulative* constraint enforces that at each point in time, the cumulated height of the set of tasks that overlap that point, does not exceed a given limit. The height of a task is its resource requirement. An *unary resource* executes one activity at a time. A *disjunctive* constraint handles a unary resource. A necessary condition for the cumulative constraint is obtained by stating a disjunctive constraint on a subset of tasks T. For each pair of tasks of T, the sum of the two corresponding minimum heights is strictly greater than the capacity. For each cumulative constraints, we add this necessary condition.
Temporal constraints, or precedences, are linear constraints between a pair of tasks. Precedences could come from constraints, decisions and propagation. Indeed, some heuristics use commitment techniques, *i.e.* assign a starting time and duration to a task, others use precedences as decisions. Propagation rules for cumulative and disjunctive could also be modified to deduce precedences.

3.2 Capacity constraints

For each route R_k, the capacity constraint is represented by a cumulative resource of capacity P_k. The requirement of T_i^k is $\sum_{j=0}^{m_i} p_i^j$. In a similar way, we add a cumulative resource of capacity $W_k \times H_k$ in which the requirement of T_i^k is $\sum_{j=0}^{m_i} w_i^j h_i^j$. This redundant constraint triggers a lower bound, proposed by Clautiaux and *al.* [7], which checks packing feasibility.

3.3 Packing constraints

Sequential loading constraint We show that the Sequential Loading Constraint (SLC) is a specialization of the non-overlapping constraint. Let m_i^s and m_j^t be two items of M. The non-overlapping constraint has the following requirements:

$$C^{NO}(m_i^s, m_j^t) \Leftrightarrow$$
$$(X_i^s + w_i^s \leq X_j^t) \vee (X_j^t + w_j^t \leq X_i^s) \vee (Y_i^s + h_i^s \leq Y_j^t) \vee (Y_j^t + h_j^t \leq Y_i^s)$$

The sequential loading constraint has the following requirements as shown in Figure 2:

$$C^{SLC}(m_i^s, m_j^t) \Leftrightarrow$$
$$\begin{cases} C^{NO}(m_i^s, m_i^t) & \text{if } s \neq t \\ (X_i^s + w_i^s \leq X_j^t) \vee (X_j^t + w_j^t \leq X_i^s) \vee \left(Y_j^t + h_j^t \leq Y_i^s\right) & \text{if } i^+ \ll j^+ \ll j^- \ll i^- \\ (X_i^s + w_i^s \leq X_j^t) \vee (X_j^t + w_j^t \leq X_i^s) & \text{if } i^+ \ll j^+ \ll i^- \ll j^- \\ \text{no constraint} & \text{otherwise} \end{cases}$$

It is straightforward to see that C^{SLC} is a restriction of C^{NO}. A global constraint is obtained by considering C^{NO} or C^{SLC} for each pair of M.

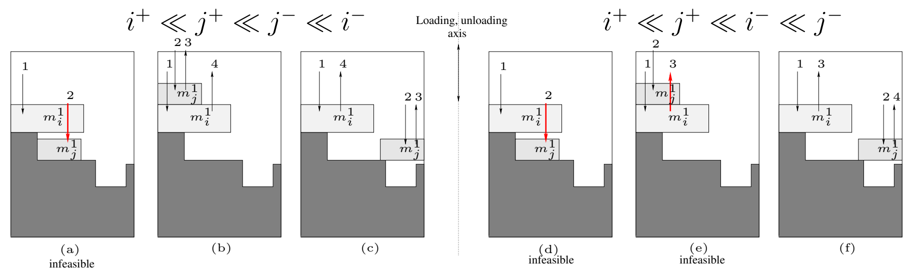

Fig. 2. Two clients i_1 and i_2 are on the same route. Each client has a demand which is a single item. We want to load and unload the items with a sequence of four authorized moves along the loading axis.

Beldiceanu and *al.* [8] describe a generic pruning technique which aggregates several constraints sharing some variables. They specialized this technique to the non-overlapping rectangles constraint. Our purpose is to specialize the algorithm to handle the sequential loading constraint instead of the non-overlapping constraint. We extended the basic filtering algorithm by changing the computation of the *forbidden region*. A forbidden region $R^{no}(m_i^s, m_j^t)$ is a set of origins of m_i^s which overlap with m_j^t. Some work is still needed in order to extend the optimized filtering algorithm. SLC also ensures that each item to be carried by the vehicle k fit completely in its loading surface.

Redundant constraints A way to enhance a CP model is to use redundant constraints, *i.e.*, a constraint which could improve the propagation at each node, although it is already ensured by the model.

A necessary condition for a feasible packing is obtained by stating a cumulative constraint on each dimension. We also can use scheduling techniques extended to packing as energetic reasoning and remaining area estimation [7]. The item's consumption is dependent on the considered node when there is pickup and delivery. If two clients i and j do not share the vehicle, *i.e.* $i^- \ll j^+ \vee j^- \ll i^+$, no constraint links their items.

4 Conclusion and perspectives

In this paper, we have proposed a scheduling based-model with a routing interface and a global constraint which handles SLC for the 2LP. We tried to apply a simple commitment heuristic inspired from the *Bottom-Left* heuristics but it is not efficient. In fact, most packing techniques use reduction procedures and symmetries to reduce the search space but these methods are incompatible with the sequential loading constraint. We want to improve our procedure by postponing the commitment phase. We consider several directions for improving our procedure: (a) searching dimension per dimension; (b) using scheduling branching schemes based on precedences; (c) reducing domains, instead of packing items, to improve the SLC propagation. (a) is equivalent to solving the cumulative problem associated to a dimension before considering the packing. (b) consists in selecting a pair of *critical tasks* and ordering them. The notion of *critical tasks* is based on the resources contention. These two schemes allow for the study of a class of solutions instead of a unique solution. This class of solutions is defined by a configuration on the first dimension or a partial order between tasks. (c) improves SLC propagation. It ensures that each item has a non-empty forbidden region before it is committed.
Later, we will use this loading model in a constructive routing heuristics and local search methods.

References

1. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. Netw. **51**(1) (2008) 4–18
2. Iori, M., Gonzalez, J.S., Vigo, D.: An exact approach for the vehicle routing problem with two dimensional loading constraints. Transportation Science **41**(2) (May 2007) 253–264
3. Ladany, S., Mehrez, A.: Optimal routing of a single vehicle with loading constraints. Transportation Planning and Technology **8** (1984) 301–306
4. Iori, M., Doener, K.F., Hartl, R.F., Fuellerer, G.: Ant colony optimization for the two-dimensional loading vehicle routing problem. Technical report, Università di Bologna (2006)
5. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search algorithm for a routing and container loading problem. Transportation Science **40**(3) (August 2006) 342–350
6. Moura, A., Oliveira, J.F.: An integrated approach to the vehicle routing and container loading problems, Greece, EURO XX - 20th Eur. Conf. on Oper. Res. (2004)
7. Clautiaux, F., Jouglet, A., Carlier, J., Moukrim, A.: A new constraint programming approach for the orthogonal packing problem. Comput. Oper. Res. **35**(3) (2008) 944–959
8. Beldiceanu, N., Carlsson, M., Thiel, S.: Sweep synchronization as a global propagation mechanism. Comput. Oper. Res. **33**(10) (2006) 2835–2851
9. Moffitt, M.D., Pollack, M.E.: Optimal rectangle packing: A Meta-CSP approach. In: Proceedings of the 16th International Conference on Automated Planning & Scheduling (ICAPS-2006). (2006)