The Storage Location Assignment and Picker Routing Problem: A Generic Branch-Cut-and-Price Algorithm

Thibault Prunet[1], Nabil Absi[1], Valeria Borodin[2], Diego Cattaruzza[3]

[1] Mines Saint-Etienne, Univ. Clermont Auvergne, INP Clermont Auvergne, CNRS UMR 6158 LIMOS, F-42023 Saint-Etienne, France
{thibault.prunet, absi}@emse.fr

[2] IMT Atlantique, LS2N, UMR CNRS 2004, Nantes, France
valeria.borodin@imt-atlantique.fr

[3] CRIStAL Centre de Recherche en Informatique Signal et Automatique de Lille, University Lille, CNRS, Centrale Lille, Inria, UMR 9189, F-59000 Lille, France
diego.cattaruzza@centralelille.fr

Keywords : *Storage location assignment, Picker routing, Column generation, e-commerce.*

1 Introduction and motivation

In supply chain management, the storage of goods in warehouses acts as a key component of the system performances [2]. The order picking activity (i.e., the action of retrieving the products from their storage locations) is largely considered the most resource-intensive activity in warehousing operations. According to [4], manual picker-to-parts warehouses, in which a human operator walks through the shelves to collect the products, were still largely dominant in 2007 and accounted for more than 80% of all warehouses in Western Europe. In such warehouses, the order picking process alone accounts for 50-75% of the total operating cost.

Three main decision problems impact the operational efficiency of order picking systems: the Storage Location Assignment Problem (SLAP), which determines an efficient assignment of Stock Keeping Units (SKUs) to storage locations, the Order Batching Problem (OBP), which groups customer orders into batches retrieved by single routes, and the Picker Routing Problem (PRP), which consists in finding the most efficient path in the warehouse to pick a given set of products from their locations. The SLAP and the PRP are strongly linked, as the PRP determines the picking routes once the product locations are known, and the quality of a SLAP solution is assessed via the computation of the routes followed by the pickers.

The SLAP, OBP and PRP have been studied extensively in the literature [4]. Most of the time, these problems are solved independently, or sequentially. Recent studies have, however, highlighted the benefits of making assignment, storage, batching and routing decisions in an integrated way in warehousing logistics [8]. The integrated optimization of the OBP and PRP is currently an active research topic [9]. The integration of the SLAP and the PRP remains marginal, as the SLAP is usually considered a tactical problem, whereas the PRP is viewed as an operational problem. However, the paradigm shift in e-commerce warehouses increases the demand variability and often requires a high level of responsiveness to ensure customer satisfaction. This translates into shorter lead times for processing and delivering incoming orders [2]. To face these challenges, a common strategy is to divide the storage area into two zones: the reserve area where products are stacked in high bays, and the smaller forward picking area where products are stored in easily reachable racks for pickers. Although the storage decisions in the reserve area remain tactical, the current trend in e-commerce warehouses is to consider them as operational in the picking area. Indeed, here the inventory quantities are small, and the picking is organized in waves, with the replenishment of depleted products generally occurring several times per day [3].

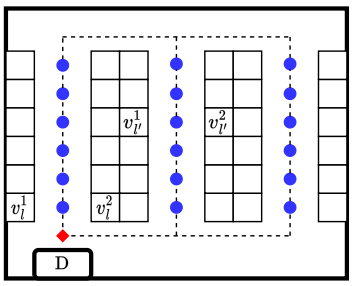

FIG. 1: Warehouse layout with storage locations (blue) and drop-off point (red).

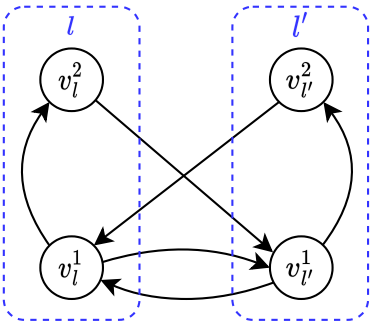

FIG. 2: Induced subgraph of $\mathcal{G}$ from nodes v_l^1, v_l^2, $v_{l'}^1$ and $v_{l'}^2$.

While a complete reoptimization of the storage plan at the operational level may seem unrealistic, the storage-location assignment decisions for replenished items are made several times per day, with complete information on the incoming demand for the upcoming cycles. This allows the explicit consideration of picker routing decisions within the problem. This dynamic storage replenishment is inspired by real-world applications of e-commerce warehouses, e.g., in Germany [10] and China [3]. Other applications involving storage decisions at an operational level include the partial reassignment of already stored items, that can happen on a daily basis in modern real-world warehouses [5].

In this context of the forward picking area, three decisions are taken before each picking cycle: (1) where to store the SKUs with insufficient inventory to meet the incoming demand (storage), (2) how to group customer orders into consolidated batches retrieved in a single route (batching) and (3) how to route the pickers (routing). Consequently, different alternatives may be considered: (a) Solve the three decision problems sequentially. (b) First, consider storage with no information on the batches. Then, jointly optimize batching and routing with perfect information on the SKU locations. (c) First, consider batching with incomplete information on the SKU locations. Then, once all batches are known, jointly optimize storage and routing with perfect information on the SKUs retrieved by each route. (d) Jointly optimize storage location, batching, and routing.

As already mentioned, the first option has been proved to lead to solutions of poor quality and is therefore discarded. In the second option, no information on the batching decisions is available to the decision-maker when deciding storage assignment, whereas in the third case, batching is determined with partial, but still rather complete information on the storage, as only a portion of the SKUs are repleted in each replenishment cycle. Finally, it is clear that a complete integration of the three problems would lead to the best results. However, we note that this integrated approach poses significant challenges and is left for future research studies.

In this paper, we assume that batching decisions are made beforehand, and we focus on the joint optimization of the storage and picker routing decisions within the forward picking area, where both decisions may be taken with detailed knowledge of the incoming demand. It is important to note that the orders we use as input data do not correspond to individual customer orders, but to consolidated batches of orders, each retrieved by a single route. The resulting problem, coined the *Storage Location Assignment and Picker Routing Problem* (SLAPRP), has been recently introduced in the literature and has been proven NP-hard in the strong sense [1].

2 Problem definition

Storage nodes and locations. We assume that all picker routes start and end at the same location v_0, called the *drop-off point*, and that a picker can indifferently pick from the right and left shelves when crossing an aisle. We call *storage node* an elementary storage space that can hold at most one SKU. We note $\mathcal{V}$ the set of storage nodes, and $\mathcal{V}^0 = \mathcal{V} \cup \{v_0\}$. We call *storage*

location a set of storage nodes that are equivalent in terms of distance from the other storage nodes. For example, in Figure 1, the storage nodes v_l^1 and v_l^2 are part of the same storage location l (represented by the blue dot in the middle). We note $\mathcal{L}$ the set of storage locations, $\mathcal{V}(l)$ the set of storage nodes that are part of location $l \in \mathcal{L}$, and $K_l = |\mathcal{V}(l)|$ its *capacity*. Note that $\mathcal{L}$ forms a partition of $\mathcal{V}$, so we can index the set as $\mathcal{V} = \bigcup_{l\in\mathcal{L}}\{v_l^1, v_l^2, \dots, v_l^{K_l}\}$. For simplicity we can use *node* (resp. *location*) instead of *storage node* (resp. *storage location*).

Graph representation. The layout of the warehouse can be modeled with a directed graph $\mathcal{G} = (\mathcal{V}^0, \mathcal{E})$ where $\mathcal{E} = \{(v_0, v_l^1) : l \in \mathcal{L}\} \cup \{(v_l^i, v_0) : l \in \mathcal{L}, i = 1, \dots, K_l\} \cup \{(v_l^i, v_{l'}^1) : l, l' \in \mathcal{L}, i = 1, \dots, K_l\} \cup \{(v_l^i, v_l^{i+1}) : l \in \mathcal{L}, i = 1, \dots, K_l - 1\}$. The graph is thus not complete: for a location $l \in \mathcal{L}$, the node v_l^1 is reachable from all the other nodes in the graph, but for $2 \leq i \leq K_l$ the node v_l^i is only reachable from node v_l^{i-1}. In other words, the node v_l^i represents the i^{th} visit of location l, which is only reachable after $(i-1)$ previous visits of l. Figure 2 provides a minimal example with two locations and four nodes. The drop-off point v_0 is reachable from all the other nodes. Note that the arcs in $\mathcal{E}$ are not associated with a weight in the general case, since the distance between two nodes depends on the routing policy. Actually, for some policies, the distance between two nodes is not constant as it depends on the nodes previously visited by a path. We assume that the distances satisfy the triangle inequality.

Problem description. The SLAPRP can be formally described as follows. A graph warehouse layout is given, represented by a graph $\mathcal{G}$ as previously introduced. The aim is to assign a set of SKUs $\mathcal{S}$ to the storage nodes $\mathcal{V}$. Without loss of generality, we assume that $|\mathcal{S}| \leq |\mathcal{V}|$. A set of orders $\mathcal{O}$ needs to be collected by the pickers. An order $o \in \mathcal{O}$ contains a subset $\mathcal{S}(o)$ of SKUs to be picked, and corresponds to one picking route. A route is defined as a tour on graph $\mathcal{G}$ starting and ending at the drop-off point. A route is valid if and only if it visits the storage nodes assigned to its set of SKUs $\mathcal{S}(o)$. Each route is associated with a cost that represents the walking distance covered by a picker and depends on the routing policy. A set $\mathcal{F} \subset \mathcal{S} \times \mathcal{L}$ of *fixed assignments* represents the locations that are already filled. For a pair $(s, l) \in \mathcal{F}$, a SLAPRP solution may only be valid if SKU s is assigned to location l. Note that $\mathcal{F}$ is empty in most variants of the SLAPRP.

3 Problem formulation

Let us introduce $\mathcal{R}_o$ as the set of possible routes to retrieve order $o \in \mathcal{O}$ and let c_{or} be the walking distance of a route $r \in \mathcal{R}_o$. We consider binary variables ρ_{or} that equal one if route $r \in \mathcal{R}_o$, is selected to collect $o \in \mathcal{O}$, 0 otherwise, and binary variables ξ_{ls} that equal one if SKU $s \in \mathcal{S}$, is placed in location $l \in \mathcal{L}$, 0 otherwise. We also consider a parameter $a_{or}^l \in \{0, 1, 2\}$ which counts the number of picks made at location l for route r to collect order o. The SLAPRP can then be formulated as follows:

$$(\mathrm{F}_{SLAPRP}) \qquad \min \sum_{o\in\mathcal{O}} \sum_{r\in\mathcal{R}_o} c_{or}\rho_{or} \tag{1}$$

$$s.t. \sum_{s\in\mathcal{S}} \xi_{ls} \leq K_l \qquad \forall l \in \mathcal{L} \tag{2}$$

$$\sum_{l\in\mathcal{L}} \xi_{ls} = 1 \qquad \forall s \in \mathcal{S} \tag{3}$$

$$\xi_{ls} = 1 \qquad \forall (s, l) \in \mathcal{F} \tag{4}$$

$$\sum_{r\in\mathcal{R}_o} \rho_{or} = 1 \qquad \forall o \in \mathcal{O} \tag{5}$$

$$\sum_{r\in\mathcal{R}_o} a_{or}^l \rho_{or} \geq \sum_{s\in\mathcal{S}(o)} \xi_{ls} \qquad \forall l \in \mathcal{L}, o \in \mathcal{O} \tag{6}$$

$$\rho_{or} \in \{0, 1\} \qquad \forall o \in \mathcal{O}, r \in \mathcal{R}_o \tag{7}$$

$$\xi_{ls} \in \{0, 1\} \qquad \forall l \in \mathcal{L}, s \in \mathcal{S} \tag{8}$$

The objective function (1) is the sum of the distances of the selected routes in the solution. Constraints (2) and (3) are the assignment constraints and ensure that the capacity of each location is satisfied and each SKU is placed in one location. Constraints (5) ensure that exactly one route is chosen to retrieve each order. Constraints (6) ensure that the route that collects $o \in \mathcal{O}$ stops at location $l \in \mathcal{L}$ if there is an SKU $s \in \mathcal{S}(o)$. They also ensure that the number of stops in this location is consistent with the number of SKUs of the order in this location.

The LP relaxation of (F_{SLAPRP}) is obtained by replacing binary requirements with non-negativity requirements on variables ρ and ξ. It is referred to as the *Master Problem* (MP). When the MP is defined over subsets of routes $\overline{\mathcal{R}_o} \subset \mathcal{R}_o$ for at least one order $o \in \mathcal{O}$ we call it the *Restricted Master Problem* (RMP).

Note that F_{SLAPRP} is *generic* in the sense that we do not assume any particular layout on the storage area, nor any particular routing strategy to collect orders.

4 Strengthened Linking Inequalities

In this section, we introduce a family of valid inequalities for F_{SLAPRP} that tighten its linear relaxation by strengthening the link between assignment variables and routing variables. The proof of validity of such inequalities is omitted due to space limitation.

Proposition 1 *Given $\overline{\mathcal{L}} \subset \mathcal{L}$, the following inequalities (coined SL inequalities) are valid for formulation F_{SLAPRP}*

$$\sum_{r \in \mathcal{R}_o} \delta_r(\overline{\mathcal{L}}) \rho_{or} \geq \sum_{l \in \overline{\mathcal{L}}} \xi_{ls} \qquad \forall o \in \mathcal{O}, s \in \mathcal{S}(o) \tag{9}$$

where $\delta_r(\overline{\mathcal{L}})$ is a parameter that equals 1 if route $r \in \mathcal{R}_o$, $o \in \mathcal{O}$, stops in one of the locations in $\overline{\mathcal{L}} \subset \mathcal{L}$, 0 otherwise.

4.1 Special case SL inequalities of order 1.

In the general case, the SL inequalities are non-robust, and their number is exponential. However, SL inequalities of order 1 (i.e., $|\overline{\mathcal{L}}| = 1$, SL-1 in the following) are robust with the proposed formulation, and their number is polynomial. Let us define the parameter b^l_{or} that equals 1 if route $r \in \mathcal{R}_o$ stops at location $l \in \mathcal{L}$, 0 otherwise. The SL-1 family of inequalities can be expressed as follows:

$$\sum_{r \in \mathcal{R}_o} b^l_{or} \rho_{or} \geq \xi_{ls} \quad \forall l \in \mathcal{L}, o \in \mathcal{O}, s \in \mathcal{S}(o) \tag{10}$$

As constraints (6) of (F_{SLAPRP}), inequalities (10) link assignment and routing variables. The difference lies in the way stops are counted: a route that stops twice in a location is counted once on the left-hand side of inequalities (10), while it is counted twice in the constraints (6).

Note that with the addition of SL-1 cuts the following theorem holds. The proof of this result is omitted due to space limitation.

Theorem 1 *Let $X = (\xi, \rho)$ be an optimal solution of the RMP, strengthened by SL-1 inequalities. If variables ξ are integral, then X is integral, or there exists an integral solution X' of the same cost.*

Note that we take advantage of the result of Theorem 1 to design an efficient branching scheme (details will be provided later).

The valid inequalities are separated using a dynamic programming (DP) procedure. The DP is solved for each $o \in \mathcal{O}$ and $s \in \mathcal{S}$ and, in order not to spend a prohibitive amount of time, several algorithmic enhancements are proposed to efficiently prune the search.

5 Branch-Cut-and-Price algorithm

To solve SLAPRP we propose a Branch-Cut-and-Price algorithm (BCP) which is a Branch-and-Bound algorithm in which the dual bound of each node of the tree is computed using the linear relaxation of an extended formulation, solved by column generation, and strengthened by the addition of cutting planes presented before.

The column generation iteratively solves a pricing problem that is in charge to find columns with a negative reduced cost that are then added to the RMP, or prove that none exists. The pricing problems for the SLAPRP are modeled as Elementary Shortest Path Problems with Resource Constraints (ESPPRC). Note that the pricing problem, and more specifically the resource extension function, is the only algorithmic component that depends on the variant considered (i.e., layout and storage policy), which makes the proposed BCP framework generic.

To recover integrality of the solution, a branching scheme is necessary. According to Theorem 1, the integrality of the ξ variables guarantees the integrality of the solution, we therefore only need to branch on such variables. Moreover, branching on the ξ variables does not impact the resolution of the subproblems. Based on these observations, a natural branching scheme, called *location branching*, would be to branch on the most fractional ξ variable. However, preliminary experiments showed that this strategy produces unbalanced search trees. To overcome this drawback, we propose a two-level branching strategy, called *combined branching*. At the first level, we branch on the assignment of SKUs in entire aisles (instead of single locations). Since integer aisle locations are not sufficient to restore integrality in the solutions, we use the location branching at the second level. All branching constraints are strengthened by a symmetry breaking procedure based on isomorphic pruning [6].

In order to speed-up the BPC, it also features, among other algorithmic enhancements, an ad-hoc primal heuristic to provide high-quality solutions, both to warm-start the BPC and provide primal bounds during the search, and a dynamic management of the active SL cuts in the formulation via a cut pool.

6 Computational experiments

The algorithm is coded in Julia 1.7.2, and CPLEX 12.10 is used to solve linear programs. We set CPLEX to use the dual simplex method, all other parameters use the default setting. The experiments are performed in *single-thread computation* on an Intel Xeon E5-2660 v3 clocked at 2.6 GHz, with a 16 GB memory limit. Each run of the BPC is limited to two hours. The detailed computational results, as well as the used instances, are available at `https://zenodo.org/record/7866860`.

Instances of [7]. These instances use a classical single block layout, with the number of aisles in $\{1, 3, 5\}$, each with $\{5, 10\}$ storage locations. The number of SKUs is between 10 and 100. The demand is characterized by $|\mathcal{O}| \in \{1, 5, 10\}$, and each order contains the same amount of products $|\mathcal{S}(o)| \in \{3, 5\}$, sampled with a uniform distribution. A total of 540 instances are solved with classical routing policies (optimal, return, S-shape, midpoint, and largest gap).

Instances of [3]. These instances are inspired by a real-life warehouse and focus on the replenishment problem. The set of fixed positions $\mathcal{F}$ is nonempty, where $\alpha \in \{20\%, 30\%, 40\%\}$ of the SKUs whose locations are free in the picking area. The demand is modeled by $|\mathcal{O}| \in \{50, 100, 200\}$, each with a random amount of SKUs in the range $|\mathcal{S}(o)| \in \{1, \ldots, 10\}$. The warehouse layout is double-block. The return routing policy is used.

Impact of the SL cuts. When F_{SLAPRP} is solved without the addition of SL cuts, it provides an average optimality gap of 7.6%. When adding the SL-1 cuts the average optimality gap goes to 6.9%. Finally, when SL-cuts are considered the average optimality gap reduces to 4.6%. Note that a compact formulation for the SLAPRP, strengthened with SL-1 cuts (which are robust and thus can be considered) provides an optimality gap of 64.6%.

Impact of the branching schemes. Results show that the combined branching scheme increases the number of solved instances and provides lower optimality gaps for the non-solved

instances. For the instances with more than one aisle, the average tree size for closed instances is reduced by 62% when combined branching is used. The gap for non-solved instances is also reduced from an average of 9.7 to 3.9 when using the combined strategy.

Overall performances. The BPC solves to optimality 115 previously unclosed instances from [7]. For the instances from [3], which were already closed, the BCP scales better with the number of orders, being on large instances several orders of magnitude faster than the previous state-of-the-art algorithm.

7 Conclusion and perspectives

In this paper, we introduced a new exact solution approach to solve the SLAPRP and a large class of its variants, including other warehouse layouts, the most common heuristic routing policies (i.e., return, S-shape, midpoint, and largest gap), and the partial replenishment variant of the problem. The developed Branch-Cut-and-Price algorithm is benchmarked on the instances from [7] and [3]. Results show that it is the new state-of-the-art algorithm for different variants of the SLAPRP.

In the future, we expect further research on integrated problems in warehousing logistics to answer the challenges of e-commerce, as it has been highlighted by other authors [2, 8]. A natural extension of the present work is the development of a matheuristic able to tackle industrial-scale instances as well as extending our algorithm to other variants of the SLAPRP (e.g., several block warehouses), which is left for future research.

References

[1] N. Boysen, and K. Stephan. The deterministic product location problem under a pick-by-order policy *Discrete Applied Mathematics*, 161(18):2862–2875, 2013.

[2] N. Boysen, and R. de Koster, F. Weidinger. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2):396–411, 2019.

[3] X. Guo, and R. Chen, and S. Du, and Y. Yu. Storage assignment for newly arrived items in forward picking areas with limited open locations *Transportation Research Part E: Logistics and Transportation Review*, 151:, 2021.

[4] R. de Koster, T. Le-Duc, K. J. Roodbergen. Design and control of warehouse order picking: A literature review *European Journal of Operational Research*, 182(2):481–501, 2007.

[5] J. Li, and M. Moghaddamand S.Y. Nof. Dynamic storage assignment with product affinity and ABC classification—a case study *The International Journal of Advanced Manufacturing Technology*, 84(9–12):2179–2194, 2016.

[6] F. Margot. Pruning by isomorphism in branch-and-cut. *Mathematical Programming*, 94(1):71–90, 2002.

[7] A. Silva, and L.C. Coelho, and M. Darvish, and J. Renaud. Integrating storage location and order picking problems in warehouse planning *Transportation Research Part E: Logistics and Transportation Review*, 140:102003, 2020.

[8] T. van Gils, and K. Ramaekers, and A. Caris, and R. de Koster Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review *European Journal of Operational Research*, 267(1):1–15, 2018.

[9] J. Wahlen, and T. Gschwind. Branch-Price-and-Cut-Based Solution of Order Batching Problems *Transportation Science*, 57(3):756–777, 2023.

[10] F. Weidinger, and N. Boysen. Scattered Storage: How to Distribute Stock Keeping Units All Around a Mixed-Shelves Warehouse *Transportation Science*, 52(6):1412–1427, 2018.