Università di Pisa

Dipartimento di Informatica

Technical Report

Sequencing and Routing in a Large Warehouse with High Degree of Product Rotation

Giacomo Lanza   Mauro Passacantando   Maria Grazia Scutellà

April 29, 2021

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy.   TEL: +39 050 2212700   FAX: +39 050 2212726

Sequencing and Routing in a Large Warehouse with High Degree of Product Rotation

Giacomo Lanza Mauro Passacantando Maria Grazia Scutellà

April 29, 2021

Abstract

The paper deals with a sequencing and routing problem originated by a real-world application context. The problem consists in defining the best sequence of locations to visit within a warehouse for the storage and/or retrieval of a given set of items during a specified time horizon, where the storage/retrieval location of an item is given. Picking and put away of items are simultaneously addressed, by also considering some specific requirements given by the layout design and operating policies which are typical in the kind of warehouses under study. Specifically, the considered sequencing policy prescribes that storage locations must be replenished or emptied one at a time by following a specified order of precedence. Moreover, two fleet of vehicles are used to perform retrieving and storing operations, whose routing is restricted to disjoint areas of the warehouse. We model the problem as a constrained multicommodity flow problem on a space-time network, and we propose a Mixed-Integer Linear Programming formulation, whose primary goal is to minimize the time traveled by the vehicles during the time horizon. Since large-size realistic instances are hardly solvable within the time limit commonly imposed in the considered application context, a matheuristic approach based on a time horizon decomposition is proposed. Finally, we provide an extensive experimental analysis aiming at identifying suitable parameter settings for the proposed approach, and testing the matheuristic on particularly hard realistic scenarios. The computational experiments show the efficacy and the efficiency of the proposed approach.

1 Introduction

In any supply chain, warehouses play a critical role. Warehousing concerns receiving, storing, order picking, and shipping of goods. In particular, order picking - the process of retrieving items from their storage locations in response to customer orders (Masae et al., 2020) - is often referred to as the most labor- and time-consuming internal logistics process. The large majority of all order picking systems is operated according to the picker-to-parts principle (especially in Western Europe according to van Gils et al., 2018), i.e., pickers walk or drive through the warehouse to retrieve products. The largest portion of an order picker's time is spent on travelling between locations. The cost of these operations is estimated to be approximately the 50% of the total operating cost of a warehouse (De Koster et al., 2007). As judged as a key point to improve productivity and decrease operational costs, the order picking problem has been widely studied in recent years (van Gils et al., 2018; Masae et al., 2020). However, in many warehouses, pickers frequently face not only the picking, but also the stocking of products. If we also consider the storage, the careful and efficient organization of workers operations becomes even greater. Nevertheless, this integrated problem has received less attention (de Brito and De Koster, 2004; Ballestín et al., 2013).

The performance of order picking and storing operations heavily depends on the locations where the goods to retrieve or store are situated or have to be situated. The possible locations for a stock keeping unit (SKU) can be broadly established from the type of *storage policy* followed in the warehouse. The most common storage policies are the dedicated storage policy, which prescribes a particular location for each SKU needing storage, the random storage policy, which involves the random assignment of SKUs to any available and eligible location within the storage area, and the class-based policy, which aims at storing groups of products at nearby positions as they are often required simultaneously (Rouwenhorst et al., 2000; De Koster et al., 2007; Gu et al., 2007). Operationally, the exact position in the warehouse is then addressed each time SKUs need to be stored, possibly subject to additional rules depending on the specific application context.

The problem we consider in this paper arises once a set of items needs to be moved towards their chosen storage locations within the warehouse (put-away operations) and a different set of items needs to be retrieved to fulfill customer order requests (picking operations). This problem is known in the literature as Sequencing and Routing Problem (SRP) and has the scope of defining the most efficient sequence of operations to move SKUs within the warehouse and perform order picking and put-away operations. The objective is typically to minimize the total material handling cost or travel efforts (measured either in time or distance traveled by the workers), while respecting some additional and peculiar requirements related to the application context (De Koster et al., 2007).

The work has been motivated by the study of a real application involving a large production site of an Italian company located in Tuscany. It is composed of a production area and a large unit-load warehouse. Its modernization is the goal of a big research project funded by Regione Toscana and it includes the resolution of a SRP for order picking and put-away operations via Operations Research techniques. The involved warehouse is larger than 10,000 m^2, has a rectangular internal layout composed of narrow storage aisles and wide cross aisles and is comprised almost entirely of storage areas. Thus, the distances traveled to perform operations are very large. SKUs are homogeneously stocked into storage locations, i.e., different types of products cannot share the same location, which is accessible only frontally from storage aisles. The warehouse relies on a random storage policy and it is characterized by a high product rotation index, i.e., more than 1,000 SKUs are moved per day. A pick-and-sort policy is also applied. Retrieved items are collected in a specific area of the warehouse (a collection area), where SKUs are sorted to establish order integrity before shipping.

The specific requirements and the warehouse design of our industrial partner allowed us to deepen some rarely discussed aspects in the literature on SRPs. Firstly, due to the particular kind of products stocked in the warehouse (tissue products for sanitary and domestic use), a strictly first-in first-out (FIFO) *sequencing policy* needs to be considered for the picking operations, prescribing that picking operations per product type must be performed by considering the time of permanence of SKUs in the warehouse. That is, oldest SKUs have to be retrieved and shipped first. This policy is largely adopted in (but not restricted to) the tissue sector to avoid the deterioration and perishability of goods (e.g., in the food sector, where items closer to their expiration date are first retrieved). As pointed out in Masae et al. (2020), the routing of pickers subject to precedence-constraints (PC) has only attracted little attention so far, without carefully considering the importance these constraints has in practice. In our context, this also implies some specific rules to consider during the sequencing of put-away operations. Specifically, the storage locations assigned to a product type have to be filled one at a time, to ease follow the FIFO policy later on during retrievals.

Moreover, two types of multi-shuttle fleets of vehicles are available to support workers in warehousing operations. However, the two types of vehicles may only travel on disjoint parts of the production site. For stock replenishment, this implies the design of a two-echelon route to move SKUs to their respective assigned storage locations, mandatorily passing through intermediate capacitated interchange points where SKUs are transferred from one vehicle type to the other one. Multiple interchange points are available within the warehouse, that is there are many alternatives where to finish the first-echelon and where to begin the second-echelon routing. Such a routing scheme is not often discussed in the literature even though it may be frequently encountered in several realistic contexts such as large end-of-line warehouses, automated storage/retrieval or human-robots shared warehouse systems. In addition, it is applied in many of the warehouses operated by our industrial partner. Some contributions discussing SRPs with different skilled fleets of vehicles have been recently considered (Ballestín et al., 2013, 2020). However, their focus is to assign (storage or retrieval) operations to the suitable skilled vehicles. To the best of the authors' knowledge, routing restrictions of vehicles within the warehouse have not been considered so far for picker-to-part warehouse systems. Additional side constraints are also considered.

We formulate the addressed SRP problem in terms of a constrained multicommodity flow problem on an time-space network, and we propose a Mixed-Integer Linear Programming (MILP) model. The SRP may be considered as a variant of the capacitated vehicle routing problem with additional constraints and it is therefore classified as NP-hard (Cuda et al., 2015; Scholz et al., 2016; Masae et al., 2020). Since real instances make the problem very hard to solve, we propose an alternative problem formulation and a matheuristic approach, based on time decomposition, which is able to solve the problem in reasonable time. Computational experiments on real-world data show the efficiency and efficacy of the proposed approach.

The main contributions of this paper can be summarized as follows: i) we study a new SRP addressing picking and put-away operations with precedence constraints and routing restrictions; ii) we provide a MILP formulation for the addressed SRP; iii) we propose a matheuristic resolution approach able to effectively determine good quality solutions to large-scale problem instances in short time; iv) we present the results of an extensive experimentation underscoring the performance of the proposed matheuristic for real large-scale instances provided by our industrial partner.

The paper is organised as follows. Section 2 reviews the main results from the literature in the area of SRP. Section 3 describes the SRP addressed in this paper. Section 4 presents the multicommodity flow based MILP formulations for the considered SRP. The matheuristic approach built to tackle the problem is presented in Section 5. Section 6 describes the experimental plan and reports the results of the computational experiments we performed. Finally, Section 7 concludes the paper and identifies some future research directions.

2 Literature Review

The SRP is an important operational problem whose aim is to define the best sequence of locations to visit within a warehouse for storing and/or retrieving a given set of items, where the storage/retrieval location of an item is given (Gu et al., 2007). In picker-to-part warehouses (the vast majority in Western Europe according to van Gils et al., 2018), operations are performed by workers who walk or drive along the aisles transporting items on vehicles, trolleys or carts. Generally, their route starts from and end in a prespecified spot within the warehouse (where they are given the list of the storage/retrieval locations to visit). The SRP normally depends on a number of warehouse-specific features such as the internal layout (e.g., length and number of aisles, presence of cross-aisles, I/O locations), the physical characteristics of the products to move (e.g., type, weight, height, shape), and the specific application context (e.g., storage policy, arrival times of products, shipment due dates, available vehicles types). Usually, the objective of SRPs relates to the optimization of travel efforts or handling times of SKUs.

Surveys on SRPs can be found in van Gils et al. (2018) and Masae et al. (2020). The authors categorized the existing literature with regard to performance measures, modelling methods and combined problems, as well as to type of algorithms (exact, heuristic, and matheuristic) and warehouse internal layout (conventional and non-conventional), respectively. Davarzani and Norrman (2015) and Gong and De Koster (2011), instead, focus on real applications and stochastic approaches, respectively. We refer to De Koster et al. (2007) and Gu et al. (2007) for a more general overview on the operational issues in warehousing problems.

More in detail, the SRP for picking activities only is a well-studied topic in the scientific literature, displaying an increasing trend of interest over the last decade (Masae et al., 2020). The most recent contributions discussing the problem focus on different realistic aspects, such as particular layout designs (Mowrey and Parikh, 2014; Scholz et al., 2016; Boysen et al., 2017; Weidinger et al., 2019; Briant et al., 2020), congestion issues (Pan and Wu, 2012; Chen et al., 2013, 2016), workers comfort (Grosse et al., 2015) and dynamic modification of list of operations (Lu et al., 2016; De Santis et al., 2018). As opposed, Gómez-Montoya et al. (2020) is the only contribution available in the literature addressing exclusively a put-away SRP.

Instead of summarizing the vast body of literature which has accumulated on these topics, we refer the interested reader to the recent above-mentioned review papers, focusing here only on those papers addressing some of the peculiarities of the SRP addressed in this work. Specifically, the main features discussed here are: *i*) the joint sequencing and routing for picking and put-away; *ii*) the use of peculiar precedence constraints in picking and storing SKUs; *iii*) the use of heterogeneous material handling equipment and the requirement of routing restrictions within the warehouse.

2.1 Joint storage and retrieval in SRP

The academic literature dealing with picker-to-part warehouse systems has focused almost exclusively on designing picking routes, making contributions focused on the combination of both picking and put-away much more scarce.

A reason is certainly due to the fact that not all picker-to-part warehouse systems are designed or operated under double command operations, i.e., where the storage needs to be planned in combination with the picking process. In some cases, in fact, the management of the two flows of items (those bounded to storage locations and those bounded instead to the output point) is actually independent, as for instance for the replenishment of a fast picking area or wave picking protocols. The double command operation certainly defines difficult larger problems to tackle, since requires the simultaneous organization of the two different flows of items, often requiring distinct management rules (Gu et al., 2007). Nevertheless, integrated schedule planning of storage/picking operations can provide the opportunity to assign resources more efficiently and have better overall performances from a practical point of view, being mentioned in several surveys as one of the topics requiring deeper attention (Davarzani and Norrman, 2015; Masae et al., 2020).

In other warehousing systems, picking and put-away are more often accounted for together, as for instance in part-to-pickers systems (where items are automatically moved between storage locations and workers by robotized or semi-robotized storage and retrieval systems, such as stacker cranes or multi-shuttles), or in specific compact-warehousing systems (such as containers in ports or steel slabs in yards). However, the considered movement restrictions of the equipment make the problem different with respect to ours. We refer the reader to the surveys of Gagliardi et al. (2012) and Carlo et al. (2014) to deepen the first and the second topic, respectively.

Focusing on picker-to-parts systems, the layout of the warehouse is a crucial aspect that may influence the performance of workers traveling from one spot to another one to fulfil storage and retrieval requests. In Pohl et al. (2009a), the efficiency of workers, measured in terms of travel time, is evaluated with respect to the three most common rectangular layouts under a single-command (dedicate a movement to either a storage or a retrieval operation) and dual-command (couple in a movement a storage and a retrieval operation together) routing protocols. In particular, their goal is to define the optimal conditions (such as size of the storage area, length of aisles, insertion of cross aisle) to apply one of the two routing protocols. Uncommon layouts (such as the Flying-V and Inverted-V aisles design) are then investigated in Pohl et al. (2009b) and Gue et al. (2012).

As a consequence of the persistent growth of the Internet as a sales channel, recent contributions of SRP for picking and put-away exclusively focus on e-commerce applications. E-commerce, in fact, allows consumers to order more products than those actually needed and return them, if not desired. Items are then shipped back to the company, and reintegrated in the stock before they are available again for reselling. Wruck et al. (2013) integrate a batching and SRP for picking and put-away, and propose multi-objective minimization models (where customers' response time and workers' travel time are minimized) for the single-worker case, considering static or dynamic creations of the list of operations to perform. In a static setting, after the list of operations is done, it is executed ignoring new storage or retrieval requests; as opposed, in a dynamic setting, re-sequence of operations is possible upon new requests. The authors propose two suitable solution approaches and validate them with a real-life case involving a library warehouse reaching an extremely high return rate. Schrotenboer et al. (2017) model a SRP for picking and put-away for a single worker as a variant of the traveling salesman problem and solve it through a genetic algorithm. A multiple-worker case is then addressed and solved by modifying single-worker routes if multiple workers interact in the same area of the warehouse. Ballestín et al. (2013) and Ballestín et al. (2020) consider SRP for picking and put-away in a chaotic warehouse where SKUs are arranged in parallel aisles composed of multi-level double-depth racks. Movements are performed by a fleet of forklifts under a single-command routing protocol. In Ballestín et al. (2013), items have associated a due date representing the time instant at which they should be retrieved at the latest not to delay trucks loading operations. They model the SRP as a project scheduling problem, whose objective is minimising the tardiness of the retrieved orders. They tested a static and a dynamic version of the problem, depending on whether the forklift driver is provided at the start of the working period with the complete list of operations to fulfil, or the list is sequentially updated with a new operation after having completed the previous one. In Ballestín et al. (2020), instead, the authors consider the problem of assigning storage locations to SKUs, selecting forklifts to perform a set of predefined picking and put-away operations, and sequencing them during the time horizon. Despite an integrated formulation is proposed, the three problems are addressed separately and sequentially.

2.2 Precedence constraints in SRP

A precedence constraint (PC) imposes that some products must be picked before some others due to some restrictions (Matusiak et al., 2014). Restrictions may vary in nature and may be related to weight or fragility issues (first retrieve heavy items), shape and size of SKUs (first retrieve big boxes), perishability (e.g., retrieving first those items closer to their expiration date), other product-category specific properties (e.g., to avoid contamination between food and non-food products) or, even, to preferred unloading sequence at customer locations. According to Masae et al. (2020) and van Gils et al. (2018), PCs in picking operations have only attracted little attention so far, even though they are encountered very often in realistic contexts. Matusiak et al. (2014) discuss a joint batching and order picking problem, where workers need to be routed in such a way that the sequence of retrievals respects a specified order related to the type (or family) of products to retrieve. The problem is modeled as a variant of the precedence-constrained travelling salesman problem (first introduced in Kubo and Kasugai, 1991). A suited heuristic solution procedure based on a decomposition approach is presented to solve real-world instances of a large Finnish warehouse. Chabot et al. (2017) describe an order picking problem with weight and product-category PCs, motivated by the practical context they examine (a grocery retail industry). They propose two mathematical models, derived from classical vehicle routing models, to formulate the problem and describe several approaches, both exact and heuristic, to solve instances with realistic size. By extending the work of Oliveira (2007), Cinar et al. (2017) consider a picking problem where items are retrieved and loaded on trucks by respecting a precedence criterion based on the order in which the clients will be visited by trucks. In particular, the last items placed in a truck are destined to the first client who will be visited. Finally, Žulj et al. (2018) consider PCs based on items weight. The authors propose a routing policy aiming at preventing product damages during retrievals. The picker handles one order per picking tour and retrieves heavy items before light ones. They propose a tour construction method that couples two subtours, the first in which only heavy items are collected and the second dedicated only to light items retrieval. The exact approach they propose evaluates all possible combinations to tie heavy and light subtours, based on the algorithm of Ratliff and Rosenthal (1983).

2.3 Multi-fleet and two-echelon SRPs

SRPs where a fleet of different skilled vehicles is available for storage and retrieval have been considered in Ballestín et al. (2013, 2020), and for retrieval only in Cortés et al. (2017). In the former, vehicles cannot access the same set of storage locations in terms of both height and depth; in the latter, vehicles have different capacities, travel speeds and lift heights. The visit to a given storage location has thus to be assigned to a suitable skilled vehicle to accomplish the task. Nevertheless, neither transshipment nor routing restrictions are considered. Vehicle routing restriction within warehouses is a very rarely topic discussed in the rich literature addressing SRPs. Oliveira (2007) and Cinar et al. (2017) consider a problem where SKUs are retrieved by automatic cranes and put on collectors where they are picked up by forklifts and loaded on trucks. A single crane operates in each aisle, and a forklift can only retrieve SKUs from its own associated collector, but deliver them to any truck. Despite the two-echelon structure of the system, the authors only focus on forklift operations, given the sequence of crane retrievals. A job-shop formulation is described and a matheuristic approach is proposed to solve real instances. Multi-echelon itineraries for items to retrieve or store are more often encountered when dealing with specific automatic parts-to-picker warehousing systems, such as for instance shuttle-based warehousing systems. They consist of two independent multi-tier subsystems, the first one controlling horizontal movements through shuttles, the second one vertical movements through a lift. All storage and retrieval operations in an aisle are completed by using both shuttles and lift. In the tier-captive configuration, each shuttle is assigned to a corresponding storage tier in one aisle. However, the number of shuttles is less than the number of storage tiers in the tier-to-tier configuration, and the shuttle itself is transferred from one storage tier to another one in the same aisle using the lift. Research on such systems is largely addressed, see for instance the recent contributions of Tappia et al. (2019); Küçükyaşar et al. (2020); Wang et al. (2020); Zhao et al. (2020).

2.4 Positioning our problem with respect to the literature

The problem considered in this paper shares some features with the problems presented in this review. Nevertheless, they have never been considered jointly in a unique setting. Firstly, regarding the precedence constraints, the above mentioned contributions only consider precedence constraints to construct picking routes. To the best of the author's knowledge, precedence constraints have never been discussed for storage operations which often need to be planned by respecting some precedence criteria. Moreover, in the literature, precedence constraints are only considered for subsets of products to pick, in particular only those included in a batch (which often correspond to a single order) entrusted to a picker. On the other hand, because of the specificity of the precedence criterion addressed in our problem, precedence constraints are applied in a broader perspective, by considering the operations of all workers and all the product types stored in the warehouse simultaneously when sequencing operations and designing routes.

Regarding fleet considerations, the literature may count on very few contributions when a not homogeneous fleet of vehicles or routing restrictions within the warehouse come into play. The latter, in particular, does not seem to have been addressed till now, except for the above mentioned unique contribution, which however focuses exclusively on the management of operations of a single part of the warehouse, while considering approximations on the operations of the other part (Cinar et al., 2017). Indeed, routing restrictions, such as for instance two-echelon routing, have been explored in contexts near SRP, particularly in the city logistics one (see for instance Crainic et al., 2009; Hemmelmayr et al., 2012). These problems, however, are different in that they do not consider demand rates, multiple vehicles per route, and multiple visits to a same location as routing workers within a warehouse normally do.

3 The problem addressed

The problem is defined in a warehouse characterized by two disjoint areas. The first area is a transit zone (e.g., a large hallway) connecting the input (or receiving) point of the warehouse - a deposit where items wait to be stored - to the storage area; the second one is the storage area, where items are stocked in storage locations. In each storage location, items are homogeneously stocked with respect to the type of product. In this area, also the output point of the warehouse is located, which is a collection area where, according to the pick-and-sort policy followed, retrieved items are gathered to establish order integrity before loading trucks. Storage locations have different capacities, depending on their location within the warehouse, and both the deposit and the collection area are capacitated.

During a specified time horizon, a number of items of different product types are placed on the deposit and require the transportation to their preassigned storage locations to replenish the stock. We define this flow of items as the *incoming flow*. At the same time, a possibly different number of items need to be picked from their storage locations and transported to the collection area to respond to the customer demands. We define this flow of items as the *outgoing flow*. Incoming items are available at the deposit at a known *availability date*, while outgoing items are required to reach the collection area before a known *due date*. The number of items and the product type of incoming and outgoing flows are known in advance and they are described in a *storing list* and a *shipping list*, respectively.

The movements of items are performed by capacitated vehicles belonging to two different types of fleets, defined in the following as F1 and F2. The routing of the two fleets of vehicles is restricted to only one of the above described disjoint areas of the warehouse. In particular, F1 can only move in the transit zone, whereas F2 can only circulate within the storage area. Vehicles may exchange freight at specific capacitated zones, called *collectors*. Items may hold on collectors with no time restrictions. Thus, incoming freight need to follow a two-echelon movement towards their storage locations. In fact, items are picked up from the deposit by a vehicle of type F1 and transported to one of the available collectors, where they are unloaded. From there, items are loaded by a vehicle of type F2 that moves them to their preassigned storage locations, where they are stored. The movement of outgoing freight is straightforward and consists of items loaded by a vehicle of type F2 from their storage locations and transported to the collection area. Nevertheless, they may idle on some collectors, once retrieved, and be transshipped from one vehicle to another one of type F2, before reaching destination. In addition, the routing of the vehicles has to be planned by considering: i) anticipation of outgoing movements with respect to the planned due dates, ii) a particular FIFO picking and put-away policy, iii) and safety requirements for workers, as better described next.

Specifically, given the high number of operations expected during each period of the planning horizon, a crucial point for the company is to anticipate some movements related to the outgoing flow of items, to ease the movements during subsequent periods. For instance, items planned to leave the site in the second period may be moved towards the collection area during the first period. This is particularly relevant when periods with a low demand are followed by periods with large demand.

Moreover, a strict management policy has to be pursued for both picking and put-away operations, separately per product type. That is, for each product type in the storing or shipping list, the operations of filling or emptying storage locations, respectively, have to follow a prespecified *order of precedence*. More in detail, for each product type in the storage list, a set of storage locations where the items have to be stored is provided alongside with the order of precedence with which such storage locations have to be filled. Consequently, separately per product type, storage locations have to be filled up one at a time following the given order of precedence, implying that a new location may be utilized for storing only if the previous one in the considered order is already completely full. This order of precedence has to be followed also when items have to be retrieved, thus generating a strict FIFO policy to follow during picking operations, again separately per product type. A motivation to consider such a retrieval order of precedence is due to the perishability of the products stored and managed in the warehouse like in the application context considered, with the need to retrieve and ship first the items of a given product type with the highest time of permanence within the warehouse.

Finally, given the very high number of movements within the warehouse and the need of multiple vehicles to work simultaneously, vehicle congestion may inevitably occur when routes are not carefully planned. Therefore, to avoid delays in warehousing operations and, most important, to guarantee security to workers, preventing congestion becomes an issue to keep in consideration. In particular, crossing and overtaking among vehicles is allowed, but no two vehicles may travel from the same location toward another same location at the same time.

4 Mathematical model

Let $\mathcal{K}$ be the set of the product types, or commodities, requiring movement in a given time horizon. The set $\mathcal{K}$ is composed of two subsets, i.e., the subset of the incoming commodities $\mathcal{K}_{in}$ and the subset of the outgoing commodities $\mathcal{K}_{out}$, (notice that $\mathcal{K}_{in}$ and $\mathcal{K}_{out}$ are not necessarily disjoint). Let $\mathcal{V}$ be the set of vehicles in charge of moving commodities inside the warehouse. It is composed of two subsets as well, i.e., the subset of vehicles belonging to fleet type F1 and the subset of vehicles belonging to fleet type F2, defined as $\mathcal{V}^1$ and $\mathcal{V}^2$, respectively.

Let $\mathcal{G}^P = (\mathcal{N}^P, \mathcal{A}^P)$ be the directed graph representing the physical network on which vehicles operate in the warehouse. Specifically, the set $\mathcal{N}^P$ defines the relevant locations of the warehouse and it includes:

- the storage locations which are pertinent to the optimization process;
- the parking areas for vehicles type F1 and F2, denoted by ω^1 and ω^2, respectively;
- the set $\mathcal{R}$ of the input points which are present in the transit area, for instance conveyors;
- the set $\mathcal{B}$ of the collectors;
- the output point (or collection area) π.

In particular, not all the storage locations of the warehouse are represented in $\mathcal{N}^P$, but only those preassigned to product types in $\mathcal{K}_{in}$, and those occupied by items of product types in $\mathcal{K}_{out}$ at the beginning of the time horizon. Hereafter, $\mathcal{S}^k_{in}$ will denote the set of storage locations in which items of product type $k \in \mathcal{K}_{in}$ have to be stored, while $\mathcal{S}^k_{out}$ will denote the set of storage locations from which items of product type $k \in \mathcal{K}_{out}$ have to be retrieved. In addition, we also define $\mathcal{S}_{in}$ as the set of all storage locations assigned to all the products $k \in \mathcal{K}_{in}$ (i.e., $\mathcal{S}_{in} = \cup_{k\in\mathcal{K}_{in}}\mathcal{S}^k_{in}$), $\mathcal{S}_{out}$ as the set of all storage locations occupied by all the products $k \in \mathcal{K}_{out}$ (i.e., $\mathcal{S}_{out} = \cup_{k\in\mathcal{K}_{out}}\mathcal{S}^k_{out}$), and $\mathcal{S}^k$ as the set of all storage locations occupied by or assigned to the products $k \in \mathcal{K}$.

As previously described, precedence relationships are associated with the set of storage locations assigned to each product type $k \in \mathcal{K}_{in}$, and with the set of storage locations occupied by each product type $k \in \mathcal{K}_{out}$, defining the order of precedence according to which storage and retrieval operations are allowed to be performed, respectively, per product type.

Regarding the set of the arcs, an arc (i, j) of $\mathcal{A}^P$ represents a direct connection between the location $i \in \mathcal{N}^P$ and the location $j \in \mathcal{N}^P$. The time to travel from i to j along (i, j), say $\tau_{i,j}$, is determined by considering the allowed speed of the vehicles and the Manhattan distance, assuming that vehicles always follow a shortest path from i to j along the network.

We model the dynamics of the problem through a *space-time network* $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. Specifically, we discretize the time horizon into T time periods of equal length through $T + 1$ time instants. The set $\mathcal{N}^P$ is then replicated $T + 1$ times, resulting in set $\mathcal{N}$. A node in $\mathcal{N}$ is defined by a couple (i, t), with $i \in \mathcal{N}^P$ and $t = 0, \ldots, T$, and represents one of the location of the warehouse at one of the considered $T + 1$ time instants. The set of arcs $\mathcal{A}$ is composed of two subsets: the subset $\mathcal{A}^H$ of the *holding* arcs and the subset $\mathcal{A}^M$ of the *moving* arcs. The subset $\mathcal{A}^H$ includes arcs of type $((i, t), (i, t + 1))$, for any $i \in \mathcal{N}^P$ and $t = 0, \ldots, T - 1$, which are used to model *idle time* of items or vehicles in a given node for one time period, while the subset $\mathcal{A}^M$ includes arcs of type $((i, t), (j, t'))$, with $i, j \in \mathcal{N}^P$, $i \neq j$ and $t, t' \in \{0, \ldots, T - 1\}$ with $t < t'$, which are used to model *movements* of items or vehicles between two different locations in different time periods. An arc $((i, t), (j, t'))$ exists in $\mathcal{A}^M$ only if in the physical network it is possible to move from $i \in \mathcal{N}^P$ to $j \in \mathcal{N}^P$. Accordingly, $t' - t = \tau_{i,j}$. We also define four subgraphs: $\mathcal{G}_{in} = (\mathcal{N}_{in}, \mathcal{A}_{in})$, $\mathcal{G}_{out} = (\mathcal{N}_{out}, \mathcal{A}_{out})$, $\mathcal{G}_{F1} = (\mathcal{N}_{F1}, \mathcal{A}_{F1})$ and $\mathcal{G}_{F2} = (\mathcal{N}_{F2}, \mathcal{A}_{F2})$, where commodities $k \in \mathcal{K}_{in}$ and $k \in \mathcal{K}_{out}$, and vehicles $v \in \mathcal{V}^1$ and $v \in \mathcal{V}^2$ may move, respectively. The complete definition of such subgraphs is given in Appendix A.

The production is defined through parameters $d^k_{in}(r, t)$, which represent the number of items of product type $k \in \mathcal{K}_{in}$ released on $r \in \mathcal{R}$ at time t to transport toward the preassigned locations, while the customers demand is defined by $d^k_{out}(\pi, t)$, which represent the number of items of product type $k \in \mathcal{K}_{out}$ requested in the collection area π at the latest time t.

A capacity is associated with each location of the warehouse: c_s represents the capacity of the storage location $s \in \mathcal{S}_{in} \cup \mathcal{S}_{out}$, c_r represents the capacity of input point $r \in \mathcal{R}$, c_π represents the capacity of the collection area π, and c_b represents the capacity of the collector $b \in \mathcal{B}$. Moreover, c_{F1} and c_{F2} represent the capacities of the vehicles type F1 and F2, respectively.

The initial state of the warehouse is defined through parameters u^k_r, u^k_b and u^k_π, for any $k \in \mathcal{K}$, $r \in \mathcal{R}$ and $b \in \mathcal{B}$, which define the number of items of product k positioned on input point r, collector b and collection area π, respectively, at the beginning of the time horizon.

In order to model the anticipation of movements of items of product types $k \in \mathcal{K}_{out}$ from the storage area towards the collection area, we need to introduce some additional parameters. The goal of such movements is to account for demands of $k \in \mathcal{K}_{out}$ beyond the considered time horizon, in order to relieve the amount of such operations in the future. We thus define an *anticipation of movements* time horizon $\tilde{T} > T$, which specifies the time periods $\tilde{t}$ beyond T, whose demand has to be preferable moved towards the collection area π before T.

Finally, we denote by $\mathcal{N}^+(i)$ and $\mathcal{N}^-(i)$ the sets of nodes linked to $i \in \mathcal{N}$ via an exiting and an entering arc, respectively, that is

$$\mathcal{N}^+(i) = \{j \in \mathcal{N} : \ \exists \ (i,j) \in \mathcal{A}\}, \qquad \mathcal{N}^-(i) = \{j \in \mathcal{N} : \ \exists \ (j,i) \in \mathcal{A}\}. \tag{1}$$

Now, let us define the four main families of variables which will be used to model the addressed SRP:

- $x^v_{(i,t)(j,t')} \in \{0,1\}$, for any $v \in \mathcal{V}^1$ and $((i,t),(j,t')) \in \mathcal{A}_{F1}$, which indicates whether vehicle v passes on the arc $((i,t),(j,t'))$, or not;
- $x^v_{(i,t)(j,t')} \in \{0,1\}$, for any $v \in \mathcal{V}^2$ and $((i,t),(j,t')) \in \mathcal{A}_{F2}$, which indicates whether vehicle v passes on the arc $((i,t),(j,t'))$, or not;
- $y^k_{(i,t)(j,t')} \in \mathbb{Z}_+$, for any $k \in \mathcal{K}_{in}$ and $((i,t),(j,t')) \in \mathcal{A}_{in}$, which indicates the number of items of product type k passing on the arc $((i,t),(j,t'))$;
- $y^k_{(i,t)(j,t')} \in \mathbb{Z}_+$, for any $k \in \mathcal{K}_{out}$ and $((i,t),(j,t')) \in \mathcal{A}_{out}$, which indicates the number of items of product type k passing on the arc $((i,t),(j,t'))$.

In addition, we introduce two families of auxiliary variables related to picking and storing policies:

- $\alpha(s^k,t) \in \{0,1\}$, for any $s^k \in \mathcal{S}^k_{in}$ and $k \in \mathcal{K}_{in}$, and $t = 0,\ldots,T$, which indicates whether the storage location s^k may be used at time t to stock product type k ($\alpha(s^k,t) = 1$), or not ($\alpha(s^k,t) = 0$);
- $\beta(s^k,t) \in \{0,1\}$, for any $s^k \in \mathcal{S}^k_{out}$ and $k \in \mathcal{K}_{out}$, and $t = 0,\ldots,T$, which indicates whether the storage location s^k may be used at time t to pick up product type k ($\beta(s^k,t) = 1$), or not ($\beta(s^k,t) = 0$).

Due to its complexity, the proposed ILP model is presented for groups of constraints, starting from the objective function. The sets, the parameters and the variables related to the model are summarized in Table 1.

Table 1: Sets, parameters and variables used in the model.

Sets	
T	no. time instants in which the time horizon is discretized
$\tilde{T}$	no. time instants for anticipation of movements
$\mathcal{K}_{in}$	set of incoming product types
$\mathcal{K}_{out}$	set of outgoing product types
$\mathcal{V}^1$	set of vehicle of fleet $F1$
$\mathcal{V}^2$	set of vehicle of fleet $F2$
ω^1, ω^2	parking areas for vehicles of $F1$ and $F2$
$\mathcal{R}$	set of input points (e.g., conveyors)
$\mathcal{B}$	set of collectors
π	collection area
$\mathcal{S}^k_{in}$	set of storage locations assigned to product type $k \in \mathcal{K}_{in}$
$\mathcal{S}^k_{out}$	set of storage locations occupied by product type $k \in \mathcal{K}_{out}$
$\mathcal{S}^k$	set of storage locations occupied/assigned to product type $k \in \mathcal{K}$
$\mathcal{G}_{in} = (\mathcal{N}_{in}, \mathcal{A}_{in})$	subgraph where product type $k \in \mathcal{K}_{in}$ may move
$\mathcal{G}_{out} = (\mathcal{N}_{out}, \mathcal{A}_{out})$	subgraph where product type $k \in \mathcal{K}_{out}$ may move
$\mathcal{G}_{F1} = (\mathcal{N}_{F1}, \mathcal{A}_{F1})$	subgraph where vehicle $v \in \mathcal{V}^1$ may move
$\mathcal{G}_{F2} = (\mathcal{N}_{F2}, \mathcal{A}_{F2})$	subgraph where vehicle $v \in \mathcal{V}^2$ may move
Parameters	
$d^k_{in}(r,t)$	no. items of product type $k \in \mathcal{K}_{in}$ released on $r \in \mathcal{R}$ at time t
$d^k_{out}(\pi,t)$	no. items of product type $k \in \mathcal{K}_{out}$ requested in π at time t
u^k_r	no. items of product type $k \in \mathcal{K}$ positioned on $r \in \mathcal{R}$ at $t = 0$
u^k_b	no. items of product type $k \in \mathcal{K}$ positioned on $b \in \mathcal{B}$ at $t = 0$
u^k_π	no. items of product type $k \in \mathcal{K}$ positioned in π at $t = 0$
c_s	capacity of storage location $s \in \mathcal{S}_{in} \cup \mathcal{S}_{out}$
c_r	capacity of $r \in \mathcal{R}$
c_π	capacity of π
c_b	capacity of $b \in \mathcal{B}$
c_{F1}, c_{F2}	capacity of $v \in \mathcal{V}^1$ or $v \in \mathcal{V}^2$
$\tau_{i,j}$	travel time between location i and j within the warehouse
Variables	
$x^v_{(i,t)(j,t')} \in \{0,1\}$	model the routing of vehicles $v \in \mathcal{V}$
$y^k_{(i,t)(j,t')} \in \mathbb{Z}_+$	model the itinerary of items of product type $k \in \mathcal{K}$
$\alpha(s^k,t) \in \{0,1\}$	model the sequencing policy for $s^k \in \mathcal{S}^k_{in}$
$\beta(s^k,t) \in \{0,1\}$	model the sequencing policy for $s^k \in \mathcal{S}^k_{out}$

Objective function

$$
\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}^1} \sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_{F1}:\\ i\neq\omega^1,\, j\neq\omega^1}} \tau_{i,j}\, x^v_{(i,t)(j,t')} + \sum_{v\in\mathcal{V}^2} \sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_{F2}:\\ i\neq\omega^2,\, j\neq\omega^2}} \tau_{i,j}\, x^v_{(i,t)(j,t')} \\
& + \psi \sum_{k\in\mathcal{K}_{in}} \sum_{\substack{((i,t),(j,t'))\in\mathcal{A}_{in}:\\ i,\, j\in\mathcal{R}}} y^k_{(i,t)(j,t')} + \xi \sum_{k\in\mathcal{K}_{out}} P^k.
\end{aligned} \tag{2}
$$

The objective function is composed of four parts. The first two summations define the primary optimization goal, i.e., minimizing the travel time of all the vehicles within the warehouse. Notice that arcs entering or leaving the parking areas are not considered for both vehicles types. This is to encourage vehicles to come back to their parking areas when idle, so limiting congestion situations along the network. The third and forth summations define soft objectives. In particular, the third summation relates to the time of permanence of the items on the input points, so as to favour the movements of items towards other spots of the warehouse. The fourth relates to the anticipation movements to perform. The latter summations are weighted through parameters ψ and ξ, respectively, to state their mutual priorities. In particular, the terms P^k are defined as follows:

$$
P^k = \max\left\{0,\ \sum_{t=0}^{\tilde{T}} d^k_{out}(\pi,t) - \left[u^k_\pi + \sum_{t=0}^{T} \sum_{(j,t')\in N^-(\pi,t)} y^k_{(j,t')(\pi,t)}\right]\right\} \tag{3}
$$

for any $k\in\mathcal{K}_{out}$. The rationale of this penalty is to compare the amount of items of type $k\in\mathcal{K}_{out}$ in the collection area, given by the last two addendum of (3) (i.e., the amount of items at the beginning of the time horizon, u^k_π, plus the items transported to π during the time horizon), with the overall demand of k, i.e., from the time instant $t=0$ to the extended time instant $\tilde{T}$, given by the first addendum of (3). The penalty term is equal to 0 if during the considered time horizon an amount of items of product k enough to satisfy the overall demand of k (i.e., in the time horizon and in the extended one) is moved to the collection area. Otherwise, the penalty to be paid is set proportionally to the amount of future demand that cannot be moved in advance.

Vehicle routing constraints

$$
\sum_{(j,t')\in\mathcal{N}^+(i,t)} x^v_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} x^v_{(j,t')(i,t)} = \begin{cases} 1 & \text{if } (i,t)=(\omega^1,\,0),\\ -1 & \text{if } (i,t)=(\omega^1,\,T),\\ 0 & \text{otherwise,}\end{cases} \tag{4}
$$
$$
\forall\ (i,t)\in\mathcal{N}_{F1},\ \forall\ v\in\mathcal{V}^1,
$$

$$
\sum_{(j,t')\in\mathcal{N}^+(i,t)} x^v_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} x^v_{(j,t')(i,t)} = \begin{cases} 1 & \text{if } (i,t)=(\omega^2,\,0),\\ -1 & \text{if } (i,t)=(\omega^2,\,T),\\ 0 & \text{otherwise,}\end{cases} \tag{5}
$$
$$
\forall\ (i,t)\in\mathcal{N}_{F2}, \forall\ v\in\mathcal{V}^2,
$$

$$
\sum_{v\in\mathcal{V}^1} x^v_{(i,t)(j,t')} \le 1 \qquad \forall\ ((i,t),(j,t'))\in\mathcal{A}_{F1}:\ i,j\neq\omega^1, \tag{6}
$$

$$\sum_{v\in\mathcal{V}^2} x^v_{(i,t)(j,t')} \leq 1 \qquad \forall\ ((i,t),(j,t')) \in \mathcal{A}_{F2}:\ i,j \neq \omega^2. \tag{7}$$

Constraints (4) and (5) ensure the correctness of the routing of the vehicles. Recalling that vehicles may only move in their respective subgraphs, (4) and (5) state that vehicles of F1 and F2 have to start their route from their parking areas (i.e., ω^1 or ω^2, respectively) at the beginning of the time horizon (i.e., at $t = 0$), and have to return there at the end of the time horizon (i.e., at $t = T$). The before mentioned security requirements are worded through constraints (6) and (7), by imposing that at most one vehicle, either of F1 or of F2, can be present in any arc of their respective subgraph. The only exceptions are for the holding arcs representing dwell time at their respective parking areas.

Incoming freight flow constraints

$$\sum_{(j,t')\in\mathcal{N}^+(i,t)} y^k_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} y^k_{(j,t')(i,t)}$$
$$= \begin{cases} d^k_{in}(i,t) + u^k_r & \text{if } i \in \mathcal{R},\ t = 0,\\ u^k_b & \text{if } i \in \mathcal{B},\ t = 0,\\ d^k_{in}(i,t) & \text{if } i \in \mathcal{R},\ t = 1,\dots,T,\\ 0 & \text{if } i \in \mathcal{B}, t = 1,\dots,T,\\ 0 & \text{if } i \in \mathcal{S}^k_{out} \cup \mathcal{S}^{k'},\ t = 0,\dots,T, \end{cases} \tag{8}$$
$$\forall\, k \in \mathcal{K}_{in},\ \forall\, k' \in \mathcal{K}:\ k' \neq k,$$
$$\forall\,(i,t) \in \mathcal{N}_{in}:\ i \in \mathcal{R} \cup \mathcal{B} \cup \mathcal{S}^k_{out} \cup \mathcal{S}^{k'}.$$

Constraints (8) are the flow conservation constraints for the incoming product types $k \in \mathcal{K}_{in}$. New releases during the time horizon are represented by values $d^k_{in}(r,t) > 0$ for some r and time instant t. For $t = 0$, it is considered the chance of already having some items idling on some r or some b, as a results of operations previously performed. Also notice that items of a product type $k \in \mathcal{K}_{in}$ can never be put in a storage location other than the one preassigned to k. The flow of each product $k \in \mathcal{K}_{in}$ thus always terminates in one of its preassigned storage locations.

Outgoing freight flow constraints

$$\sum_{(j,t')\in\mathcal{N}^+(i,t)} y^k_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} y^k_{(j,t')(i,t)} = \begin{cases} u^k_b & \text{if } i \in \mathcal{B}, t = 0,\\ 0 & \text{if } i \in \mathcal{B}, t \geq 1,\\ 0 & \text{if } i \in \mathcal{S}^k_{out} \cup \mathcal{S}^{k'} \\ & \text{and } t \geq 0, \end{cases} \tag{9}$$
$$\forall\ k \in \mathcal{K}_{out},\ \forall\ k' \in \mathcal{K}:\ k \neq k',$$
$$\forall\ (i,t) \in \mathcal{N}_{out}:\ i \in \mathcal{B} \cup \mathcal{S}^{k'} \cup \mathcal{S}^k_{in},$$

$$\sum_{(j,t')\in\mathcal{N}^-(\pi,t)} y^k_{(j,t')(\pi,t)} - \sum_{(\pi,t')\in\mathcal{N}^+(\pi,t)} y^k_{(\pi,t)(\pi,t')} = d^k_{out}(\pi,t) \tag{10}$$
$$\forall\ k \in \mathcal{K}_{out},\ \forall\ t \geq 1,$$

$$y^k_{(\pi,0)(\pi,1)} = u^k_\pi \qquad \forall\, k \in \mathcal{K}_{out}. \tag{11}$$

Relations (9) are the flow conservation constraints for $k \in \mathcal{K}_{out}$. As for the case of the incoming freight flow, it is considered the chance of having some items of product type $k \in \mathcal{K}_{out}$ idling on some b at time $t = 0$, as a result of operations previously performed. Moreover, items of a product type $k \in \mathcal{K}_{out}$ can never be stored in any storage location once retrieved. Relations (10)– (11) are demand constraints. In particular, constraints (10) ensure that all the items of product type $k \in \mathcal{K}_{out}$ requested at time t are transported to the collection area before t, while (11) defines the composition of the collection area at the beginning of the time horizon.

Linking capacity constraints

$$\sum_{\substack{k \in \mathcal{K}_{in}: \\ ((i,t),(j,t')) \in \mathcal{A}_{in}}} y^k_{(i,t)(j,t')} \le c_{F1} \sum_{v \in \mathcal{V}^1} x^v_{(i,t)(j,t')} \qquad \forall\, ((i,t),(j,t')) \in \mathcal{A}_{F1}, \tag{12}$$

$$\sum_{\substack{k \in \mathcal{K}_{in}: \\ ((i,t),(j,t')) \in \mathcal{A}_{in}}} y^k_{(i,t)(j,t')} + \sum_{\substack{k \in \mathcal{K}_{out}: \\ ((i,t),(j,t')) \in \mathcal{A}_{out}}} y^k_{(i,t)(j,t')} \le c_{F2} \sum_{v \in \mathcal{V}^2} x^v_{(i,t)(j,t')} \qquad \forall\, ((i,t),(j,t')) \in \mathcal{A}_{F2}. \tag{13}$$

Relations (12)–(13) define the linking capacity constraints for vehicles of F1 and F2, respectively, by considering both incoming and outgoing items flows. In particular, they state that freight flows can only be transported by means of vehicles which have been selected to move within the warehouse, and that the total commodity flow on any moving arc cannot exceed the capacity of the vehicle traveling along it.

Location capacity constraints

$$\sum_{k \in \mathcal{K}_{in}} d^k_{in}(r,t) + \sum_{k \in \mathcal{K}_{in}} y^k_{(r,t-1)(r,t)} \le \begin{cases} c_r - \sum_{k \in \mathcal{K}_{in}} u^k_r & \text{if } t = 1, \\ c_r & \text{if } t > 1, \end{cases} \qquad \forall\, r \in \mathcal{R},\ \forall\, t \ge 1, \tag{14}$$

$$\sum_{(j,t') \in \mathcal{N}^-(b,t)} \sum_{k \in \mathcal{K}} y^k_{(j,t')(b,t)} \le \begin{cases} c_b - \sum_{k \in \mathcal{K}} u^k_b & \text{if } t = 1, \\ c_b & \text{if } t > 1, \end{cases} \qquad \forall\, b \in \mathcal{B},\ \forall\, t \ge 1, \tag{15}$$

$$\sum_{(j,t') \in \mathcal{N}^-(\pi,t)} \sum_{k \in \mathcal{K}_{out}} y^k_{(j,t')(\pi,t)} \le \begin{cases} c_\pi - \sum_{k \in \mathcal{K}_{out}} u^k_\pi & \text{if } t = 1, \\ c_\pi & \text{if } t > 1, \end{cases} \qquad \forall\, t \ge 1, \tag{16}$$

$$\sum_{\tilde{t}=0}^{t} \left[\sum_{(j,t') \in \mathcal{N}^-(i,\tilde{t})} y^k_{(j,t')(i,\tilde{t})} - \sum_{(j,t') \in \mathcal{N}^+(i,\tilde{t})} y^k_{(i,\tilde{t})(j,t')} \right] \le c_i \qquad \forall\, k \in \mathcal{K}_{in},\ \forall\, t \ge 1,\ \forall\, (i,t) \in \mathcal{N}_{in} :\ i \in \mathcal{S}^k_{in}, \tag{17}$$

$$\sum_{\tilde{t}=0}^{t}\left[\sum_{(j,t')\in\mathcal{N}^{+}(i,\tilde{t})} y^{k}_{(i,\tilde{t})(j,t')} - \sum_{(j,t')\in\mathcal{N}^{-}(i,\tilde{t})} y^{k}_{(j,t')(i,\tilde{t})}\right] \leq c_i \tag{18}$$
$$\forall\, k \in \mathcal{K}_{out},\ \forall\, t \geq 1,\ \forall\, (i,t) \in \mathcal{N}_{out}:\ i \in \mathcal{S}^{k}_{out}.$$

Relations (14)–(18) define the capacity constraints for each location of the warehouse. In particular, constraints (14) relate to input points, constraints (15) relate to collectors, and constraints (16) relate to the collection area. Moreover, by considering the incoming flow, constraints (17) guarantee the satisfaction of the capacity of each storage location preassigned to $k \in \mathcal{K}_{in}$. Similarly, by considering the outgoing flow, constraints (18) state the maximum number of items that can be retrieved from storage locations occupied by product types $k \in \mathcal{K}_{out}$.

Storage policy constraints

$$\sigma^{t}_{s^k} = \sum_{\tilde{t}=0}^{t}\left[\sum_{(j,t')\in\mathcal{N}^{-}(s^k,\tilde{t})} y^{k}_{(j,t')(s^k,\tilde{t})} - \sum_{(j,t')\in\mathcal{N}^{+}(s^k,\tilde{t})} y^{k}_{(s^k,\tilde{t})(j,t')}\right] \tag{19}$$
$$\forall\, k \in \mathcal{K}_{in},\ \forall\, s^k \in \mathcal{S}^{k}_{in},\ \forall\, t \geq 1,$$

$$c_{s^k_l} - \sigma^{t}_{s^k_l} \leq c_{s^k_l}\left[1 - \alpha(s^k_{l+1}, t)\right] \qquad \forall\, k \in \mathcal{K}_{in},\ \forall\, t = 0,\dots,T,\ \forall\, s^k_l \in \mathcal{S}^{k}_{in},\ \forall\, l = 1,\dots,|\mathcal{S}^{k}_{in}| - 1, \tag{20}$$

$$\sum_{(j,t')\in\mathcal{N}^{-}(s^k,t)} y^{k}_{(j,t')(s^k,t)} \leq c_{s^k}\, \alpha(s^k, t) \qquad \forall\, k \in \mathcal{K}_{in},\ \forall\, s^k \in \mathcal{S}^{k}_{in},\ \forall\, t \geq 0. \tag{21}$$

In accordance with the specific storage policy, storage locations are required to be filled sequentially by respecting the specified order of precedence. For example, let $s^k_1 \in \mathcal{S}^k_{in}$ be the first storage location eligible for stocking the product type $k \in \mathcal{K}_{in}$ and $s^k_2 \in \mathcal{S}^k_{in}$ be the second storage location eligible for stocking (in accordance to the order of precedence of the preassigned storage locations). At the beginning of the time horizon, i.e., at $t = 0$, stocking has to begin from s^k_1 and then continue, only once it is full, by using the next preassigned storage location, i.e., s^k_2. Constraints (19)–(21) state this policy. In particular, equations (19) define the total number of items of product type $k \in \mathcal{K}_{in}$ stocked in the storage location $s^k \in \mathcal{S}^k_{in}$ until time t (note that, at $t = 0$ and for the first storage location in the given order of precedence, this is an input data). If storage location s^k_l has not already reached its saturation at time t, constraints (20) do not allow the next assigned storage location in the related order of precedence, i.e., s^k_{l+1}, to be used to stock items of product type $k \in \mathcal{K}_{in}$: this is mathematically guaranteed by forcing $\alpha(s^k_{l+1}, t) = 0$ in this scenario thanks to constraints (20). As opposed, when storage location s^k_l has reached its saturation, i.e., $c_{s^k_l} = \sigma^t_{s^k_l}$, storage location s^k_{l+1} becomes eligible to stock items of product type k, being $\alpha(s^k_{l+1}, t)$ allowed by the combination of constraints (20) and (21) to assume value 1, that is $\alpha(s^k_{l+1}, t) = 1$.

Retrieval policy constraints

$$\rho^{t}_{s^k} = \sum_{\tilde{t}=0}^{t}\left[\sum_{(j,t')\in\mathcal{N}^{+}(s^k,\tilde{t})} y^{k}_{(s^k,\tilde{t})(j,t')} - \sum_{(j,t')\in\mathcal{N}^{-}(s^k,\tilde{t})} y^{k}_{(j,t')(s^k,\tilde{t})}\right] \tag{22}$$
$$\forall\, k \in \mathcal{K}_{out},\ \forall\, s^k \in \mathcal{S}^{k}_{out},\ \forall\, t \geq 1,$$

$$c_{s_l^k} - \rho^t_{s_l^k} \leq c_{s_l^k}\,(1 - \beta(s^k_{l+1}, t)) \qquad \begin{array}{l} \forall\, k \in \mathcal{K}_{out},\ \ \forall\, t \geq 0, \\ \forall\, s^k_l \in \mathcal{S}^k_{out},\ \ \forall\, l = 1, \ldots, |\mathcal{S}^k_{out}| - 1, \end{array} \tag{23}$$

$$\sum_{(j,t') \in \mathcal{N}^+(s^k,t)} y^k_{(s^k,t)(j,t')} \leq c_s\,\beta(s^k, t) \qquad \begin{array}{l} \forall\, k \in \mathcal{K}_{out},\ \ \forall\, s^k \in \mathcal{S}^k_{out}, \\ \forall\, t \geq 0. \end{array} \tag{24}$$

In accordance with the specific retrieval policy, storage locations are required to be emptied sequentially by respecting their specified order of precedence. Constraints (22)–(24), whose logic is similar to constraints (19)–(21), state this policy. In particular, equations (22) define the total number of items of product type $k \in \mathcal{K}_{out}$ retrieved from the storage location $s^k \in \mathcal{S}^k_{out}$ until time t (also in this case, at $t = 0$ and for the first storage location in the given order of precedence, this is an input data). Constraints (23) impose that the next storage location in the related order of precedence, s^k_{l+1}, cannot be used to retrieve items of product type k, unless the previous storage location, s^k_l, has been completely emptied. In the latter case, $\beta(s^k_{l+1}, t) = 1$ is allowed by the combination of constraints (23) and 24; otherwise $\beta(s^k_{l+1}, t) = 0$ and retrieval still has to be performed from s^k_l.

4.1 Enhanced formulation

The dimension of the model presented in Section 4 may rapidly raise as the number of the storage locations pertinent to the optimization process increases. Therefore, we consider also an alternative formulation defined on an alternative graph, whose nodes are not associated with the individual storage locations (as described in Section 4), rather they are associated with groups of contiguous storage locations, with sequential priority, which are either occupied or assigned for storing to a same product type. In the following, we refer to such a group of contiguous storage locations as a *super-storage location* (SSL for short). The capacity of a SSL is the sum of the capacities of the single storage locations composing it. Let $\tilde{\mathcal{S}}^k_{in}$ be the set of SSLs assigned to a product type $k \in \mathcal{K}_{in}$ for storing operations, let $\tilde{\mathcal{S}}^k_{out}$ be the set of SSLs occupied by a product type $k \in \mathcal{K}_{out}$, and let $\tilde{\mathcal{S}}_{in}$ and $\tilde{\mathcal{S}}_{out}$ be respectively the set of all SSLs assigned to all the products $k \in \mathcal{K}_{in}$ and the set of all SSLs occupied by all the products $k \in \mathcal{K}_{out}$, we define the capacity of SSL $\tilde{s} \in \tilde{\mathcal{S}}_{in} \cup \tilde{\mathcal{S}}_{out}$ as $\tilde{c}_{\tilde{s}}$. Using the notation introduced above, the SSL formulation can be obtained by appropriately replace sets and parameters associated to storage locations with those associated to SSL in model (2)–(24).

Notice that, if on the one hand the alternative representation of the storage locations of the warehouse in terms of SSL may bring to a reduction of the dimension of the associated graph, and thus ease the resolution process, on the other hand this may lead to a less manageable solution, since workers have now information about storing or picking operations not at a storage location level but rather at a SSL level.

5 Matheuristic approach

For real instances, such as those provided to us by our industrial partner, the proposed formulations may have a very high dimension because of the huge number of products and storage locations involved in storing and retrieving operations (recall that we address warehouses with a high degree of product rotation). Thus the models cannot be directly addressed through the state-of-the-art commercial solver CPLEX. Therefore, we propose a matheuristic approach based on a decomposition strategy. Specifically, the planning horizon is divided into Λ subperiods, by splitting the original time horizon into Λ periods of equal (or different) length. Each subperiod thus gives rise to a subproblem, whose features are those of the original problem restricted to the considered subperiod. The Λ subproblems are then sequentially solved by using CPLEX, in such a way that the final state of the system obtained solving subproblem $\lambda - 1$ becomes the initial state of the system for solving subproblem λ, for any $\lambda = 2, \ldots, \Lambda$. In particular, the state of the system considers the position of vehicles and items within the warehouse. Finally, the obtained Λ solutions are sequentially unified to define the solution of the original problem.

The subperiod reformulation may be derived straightforwardly from the complete planning horizon formulation described in Section 4, by keeping unchanged the structure of the majority of its constraints. Nevertheless, parameter T now defines the final time instant of the generic subperiod, instead of the end of the whole time horizon. Moreover, constraints (4), (5), (8) and (9) needs to be modified. Specifically, since in subperiod $\lambda = 1, \ldots, \Lambda - 1$, the vehicles of F1 are not obliged to go back to their parking area at the end of it, i.e., at time T, and defining as $u^v \in \mathcal{N}_{F1}$ the physical node from where the vehicle $v \in \mathcal{V}^1$ begins its route in subperiod $\lambda > 1$, the set of constraints (4) is modified in the following way:

- if $\lambda = 1$, then the vehicles depart from their parking area:

$$\sum_{(j,t') \in \mathcal{N}^+(i,t)} x^v_{(i,t)(j,t')} - \sum_{(j,t') \in \mathcal{N}^-(i,t)} x^v_{(j,t')(i,t)} = \begin{cases} 1 & \text{if } (i,t) = (\omega^1, 0), \\ 0 & \text{otherwise}, \end{cases} \quad \forall\, v \in \mathcal{V}^1,\ \forall\, (i,t) \in \mathcal{N}_{F1}; \tag{25}$$

- if $\lambda = \Lambda$, then the vehicle v departs from u^v and then returns to the parking area at the end of the subperiod:

$$\sum_{(j,t') \in \mathcal{N}^+(i,t)} x^v_{(i,t)(j,t')} - \sum_{(j,t') \in \mathcal{N}^-(i,t)} x^v_{(j,t')(i,t)} = \begin{cases} 1 & \text{if } (i,t) = (u^v, 0), \\ -1 & \text{if } (i,t) = (\omega^1, T), \\ 0 & \text{otherwise}, \end{cases} \quad \forall\, v \in \mathcal{V}^1,\ \forall\, (i,t) \in \mathcal{N}_{F1}; \tag{26}$$

- if $1 < \lambda < \Lambda$, then the vehicle v starts its route from the node where it ended in the previous subperiod, i.e., u^v:

$$\sum_{(j,t') \in \mathcal{N}^+(i,t)} x^v_{(i,t)(j,t')} - \sum_{(j,t') \in \mathcal{N}^-(i,t)} x^v_{(j,t')(i,t)} = \begin{cases} 1 & \text{if } (i,t) = (u^v, 0), \\ 0 & \text{otherwise}, \end{cases} \quad \forall\, v \in \mathcal{V}^1,\ \forall\, (i,t) \in \mathcal{N}_{F1}. \tag{27}$$

The same applies to vehicles of fleet F2, therefore constraints (5) are similarly modified.

In addition, it needs to be considered that at the beginning of a subperiod λ some items of $k \in \mathcal{K}$ may be in front of a storage location to which it is not assigned (just passing) as a result of operations in subperiod $\lambda - 1$. Let u_s^k be the number of items of product $k \in \mathcal{K}$ located in front of a storage location $\mathcal{S}_{in} \cup \mathcal{S}_{out}$ at the beginning of subperiod λ. Constraints (8) and (9) are modified as follows:

- for product types in $\mathcal{K}_{in}$:

$$\sum_{(j,t')\in\mathcal{N}^+(i,t)} y^k_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} y^k_{(j,t')(i,t)} = \begin{cases} d^k_{in}(i,t) + u^k_r & \text{if } i \in \mathcal{R},\ t = 0, \\ d^k_{in}(i,t) & \text{if } i \in \mathcal{R},\ t = 1, \dots, T, \\ u^k_b & \text{if } i \in \mathcal{B},\ t = 0, \\ u^k_s & \text{if } i \in \mathcal{S}^k_{out} \cup \mathcal{S}^{k'},\ t = 0, \\ 0 & \text{if } i \in \mathcal{B} \cup \mathcal{S}^k_{out} \cup \mathcal{S}^{k'},\ t = 1, \dots, T, \end{cases} \tag{28}$$

$$\forall\ k \in \mathcal{K}_{in},\ \forall\ k' \in \mathcal{K}:\ k' \neq k,$$
$$\forall\ (i,t) \in \mathcal{N}_{in}:\ i \in \mathcal{R} \cup \mathcal{B} \cup \mathcal{S}^k_{out} \cup \mathcal{S}^{k'};$$

- for product types in $\mathcal{K}_{out}$:

$$\sum_{(j,t')\in\mathcal{N}^+(i,t)} y^k_{(i,t)(j,t')} - \sum_{(j,t')\in\mathcal{N}^-(i,t)} y^k_{(j,t')(i,t)} = \begin{cases} u^k_b & \text{if } i \in \mathcal{B},\ t = 0, \\ u^k_s & \text{if } i \in \mathcal{S}^k_{out} \cup \mathcal{S}^{k'},\ t = 0, \\ 0 & \text{if } i \in \mathcal{B} \cup \mathcal{S}^k_{out} \cup \mathcal{S}^{k'},\ t = 1, \dots, T, \end{cases} \tag{29}$$

$$\forall\, k \in \mathcal{K}_{out}, \forall\, k' \in \mathcal{K}:\ k' \neq k,$$
$$\forall\, (i,t) \in \mathcal{N}_{out} : i \in \mathcal{B} \cup \mathcal{S}^{k'} \cup \mathcal{S}^k_{in}.$$

The matheuristic approach is summarized in Algorithm 1.

Algorithm 1 The matheuristic approach

1: Divide the time horizon T into Λ time periods
2: $\Phi_0 = \emptyset$
3: **for** $\lambda = 1, \dots, \Lambda$ **do**
4:     Solve the λ-th subproblem
5:     Save the solution of the λ-th subproblem as Φ_λ
6: **end for**
7: Unify the subproblem solutions: $\Phi = \cup_\lambda \Phi_\lambda$

6 Numerical experiments

6.1 The case study addressed

The production site of the company we consider, leader in the tissue sector, is composed of a production area, a large warehouse, a collection area, and several shipping docks. The warehouse is larger than 10,000 m^2 and it is located beside the production area and connected to it by a large hallway. The warehouse is composed of 4 departments. Each department has a rectangular internal layout with a certain number of parallel narrow storage aisles and parallel wide cross aisles. The storage area is thus divided in blocks of storage locations framed by aisles. Items are homogeneously (with respect to the product type) stored back-to-back to each other in each storage location, in such a way to define horizontal stacks of items of the same type, accessible only frontally. A random storage policy (respecting though the homogeneity criteria) is applied. Different blocks may be composed of different number of stacks, all having though the same capacity. However, stacks belonging to different blocks may have different capacities. Specifically, the storage area is divided into 29 blocks, which are composed of a variable number of stacks ranging from 15 to 65. Stacks have a capacity ranging from 8 to 17 items, independently on the product type to store. According to the pick-and-sort policy followed, the collection area is used to gather retrieved items and establish order integrity before loading the trucks, and it is positioned at the end of the forth department. It can stock up to 700 items, and is normally filled up as much as possible during the night to quickly start the truck loading operations the next morning.

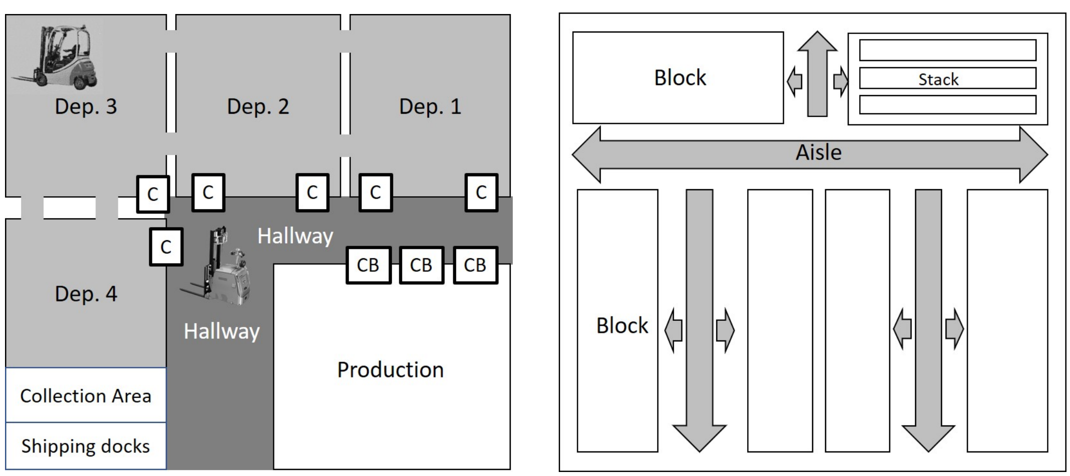
The production site works daily on 3 shifts of 8 hours. Production never stops during the day, while orders are shipped during the first and the second shift only. More than 300 different types of products are produced in this site. Items are released by the production on 3 end-of-line conveyor belts (just conveyors in the following), arranged in unit-loads and wrapped in so-called *columns* of pallets. Therefore, the inventory will be expressed in terms of columns in our study. Conveyors can hold a limited quantity of columns (precisely, 10, 14 and 8 columns, respectively) and need to be emptied as soon as possible when columns are released not to block subsequent releases (production decisions are independent, and they are not addressed here). Items are released at a constant rate during the shift. Each release is characterized by a *release time instant*, an *amount of columns* released per product type, and the conveyor of release. Additionally, the storing list also reports, separately per product type, the set of assigned stacks to utilize to store that product type, and the order of precedence in which they have to be filled up. Assignment storage location decisions are not part of this study, and are discussed in Lanza et al. (2021). The shipping list of a day is normally known a day in advance and reports the composition of an order, that is amount of columns and types of product requested, and the leaving time of the associated truck. Items are required to be retrieved from stacks following the given order of precedence per product type, and they are moved to the collection area before a given *due date*, not to generate truck loading delay.

The fleet of the company is composed of 5 LGV shuttles and 7 forklifts (LGV and FKL in the following). Referring to the more general problem description in Section 3, LGV and FKL correspond to vehicles of type F1 and F2, respectively. Both types of vehicles may transport two columns at most at the same time. LGV may only move on the hallway connecting conveyors and departments, while FKL may move within the departments and the collection area. Collectors are positioned at the entrance of each department. The number of collectors within the warehouse is 6, with different capacities ranging from 2 to 8 columns. Items may hold on collectors with no time restrictions, but generally it is preferable to move them as soon as possible towards their destination, be this a storage location or the collection area, no as to generate congestion of items around the warehouse. Incoming items are thus moved from conveyors to collectors by LGV, possibly idling on collectors, and then moved from collectors to stacks by FKL. Outgoing items, instead, are moved from stacks to the collection area by FKL, by possibly idling on collectors as well. LGV and FKL are allowed to cross and overtake each other in their respective routing areas, but no two vehicles may travel from the same location toward another same location at the same time, to limit the congestion.

Moreover, given the high number of operations required during each shift, a crucial point for the company is to anticipate as much as possible the movements of requested items towards the collection area during a shift, to ease the work load during the subsequent shift. So, for instance, items planned to leave the site during the second shift of a day, may be moved towards the collection area already during the first shift. This is particularly needed for the third shift, where the collection area is filled up as much as possible to quickly load trucks the next morning.

As in the general presentation in Section 3, critical issues are thus to perform storage and retrieval operations by following a strict order of precedence, to avoid vehicle congestion, and to anticipate movements for outgoing items during each shift.

The structure of the warehouse is depicted in Figure 1a. The positions of conveyor belts (denoted with CB) and of collectors (denoted with C) are also reported. The areas of the warehouse are filled with different colors, namely dark grey and light grey, indicating the areas where LGV and FKL are allowed to move, respectively. Figure 1b shows the internal structure of a department.



(a) Warehouse representation (b) Department representation

Figure 1: Warehouse and department representations

6.2 Plan of the experiments

Two types of experiments have been performed. In Section 6.4, we analyze the performance of the matheuristic approach described in Section 5 by considering both formulations previously introduced. In the following, we will refer to the storage-location-based formulation described in Section 4 and to the super-storage-location-based formulation described in Section 4.1 as SL and SSL, respectively. The efficacy and the efficiency of the approach, when using either SL or SSL, have been tested on a wide pool of real instances related to the addressed case study, with the aim of identifying suitable parameter settings for the proposed approach. The selected time horizon is a shift (8 hours). The instances are described in Section 6.3.

Then, in Section 6.5 we further investigate the efficiency and the efficacy of the approach by considering as input data one of the busiest weeks for our industrial partner, where the movements of items are far beyond the annual average. This analysis involves the consecutive resolution of the addressed SRP problem for each day of the selected week, by considering the formulation and the setting of the parameters suggested by the first type of experiment.

The matheuristic approach has been implemented using the OPL language and solved via CPLEX 12.6 solver (IBM ILOG, 2016). All the experiments have been conducted on an Intel Xeon 5120 computer with 2.20 GHz and 32 GB of RAM.

6.3 The instances

The data set provided by the company comprises the following information for a pool of selected shifts:

(i) the warehouse configuration at the beginning of the shift, i.e., product types and corresponding number of columns inside the warehouse;

(ii) the storing list of the shift;

(iii) the shipping list of the shift and of the next three shifts.

Some data needed to be integrated, others instead, not provided by the company, were randomly generated. Specifically, the positions of the columns in the warehouse at the beginning of a shift are randomly generated by respecting some agreed industrial practice or insights given by the company, in such a way as to start with a realistic configuration. Additionally, the retrieval order of precedence per product type for the occupied stacks in the warehouse needed to be randomly generated as well. On the other hand, the stacks assigned to each product type in the storing list and the corresponding filling order of precedence have been obtained by applying the model and the resolution method proposed in Lanza et al. (2021). Finally, the truck leaving times have been randomly generated by considering that the majority of the orders are shipped during the morning.

For the first type of analysis, 15 shifts have been selected, by thus generating 15 corresponding instances. The second type of analysis, instead, has been performed on one of the busiest weeks for our industrial partner, as previously outlined.

6.4 Efficacy and efficiency of the matheuristic approach

The matheuristic approach relies either on the SL or on the SSL formulation, and it is characterized by some parameters.

As described in Section 5, the matheuristic consists of a decomposition of the time horizon, i.e., a shift, for which the resolution process via CPLEX often is not addressable, into smaller periods, i.e., subshifts, for which instead CPLEX easily finds solutions in reasonable time. The time length of a subshift is a key parameter of the approach. We tested three different time lengths. Recalling that a shift lasts 8 hours, we split the time horizon into 10, 16 and 30 subshifts. This corresponds to have subshifts of about 50, 30 and 15 minutes, respectively. Moreover, two different values for parameters ψ and ξ are tested. Increasing values of ψ would tend to give priority to emptying conveyors, moving columns as soon as they are released from the production area towards the collectors. Increasing values of ξ, instead, would tend to give priority to the anticipation movements toward the collection area. After some preliminary tests, we decided to analyse values 10 and 50 for both ψ and ξ, in their four combinations. In the following, we refer to a weight combination as a pair of numbers in brackets of type $(\cdot - \cdot)$, where the first position is associated with ψ and the second with ξ. The tested weight combinations are thus (10-10), (10-50), (50-10) and (50-50).

Therefore, for both the variants SL and SSL, the three time settings related to the time length of a subshift and the four weight combinations of ψ and ξ have been combined. Each of the 15 instances composing the data set has thus been solved 12 times by considering the SL formulation (180 runs) and 12 times by considering the SSL formulation (180 runs), for a total of 360 runs.

The time limit required by the company to obtain solutions is 240 minutes for an entire shift. In our resolution approach, the total time needed to obtain a solution for the entire shift is the sum of the time needed to solve each of the subproblems in which the shift is split. We thus imposed a different time limit on the resolution of the subproblems depending on whether 10, 16 or 30 subproblems are considered. Specifically, if 10 subproblems are considered, the time limit imposed to each subproblem is 24 minutes; if 16 subproblems are considered, the time limit imposed to each subproblem is 15 minutes; finally, if 30 subproblems are considered, the time limit imposed to each subproblem is 8 minutes. The algorithm may stop the resolution of a subproblem before reaching the associated time limit, if the estimated percentage gap between the optimum and the current solution value is lower than 10%.

We firstly investigated the impact of the used formulation (SL or SSL) on the efficiency and efficacy of the matheuristic approach by comparing the total number of instances the approach is able to solve within the time limit imposed. The latter are reported in Table 2 for the alternative formulations SL and SSL, the three selected time lengths for subshifts and the 4 combinations of weights ψ and ξ. The former are indicated by NS-(number of total subshifts), while the latter are reported on the first and second columns in the table.

Table 2: Number of instances solved by the resolution approach

Weights		SL formulation			SSL formulation		
ψ	ξ	NS-10	NS-16	NS-30	NS-10	NS-16	NS-30
10	10	3	7	13	6	15	15
50	10	3	7	13	6	15	15
10	50	1	3	4	6	7	9
50	50	3	6	12	7	15	15

Despite the reduction in problem size led by the proposed time horizon decomposition, when the SL formulation is considered, the matheuristic seems to still generate too large subproblems that CPLEX is hardly able to solve. CPLEX in fact finds solutions only to the minority of the tested instances. The finer time horizon decomposition, namely NS-30, seems to be the most suitable algorithm setting in this case, even though not all the instances are successfully solved. Interestingly, the combination of weights $\psi = 10$ and $\xi = 50$ seems to generate the hardest subproblems than any other combination of weight values. This may be explained by considering that, in any instance, the number of items requiring movements towards stacks is lower than the number of items requiring movements towards the collection area. The latter, in fact, is associated with the present and the future demand to satisfy due to the anticipation movements policy considered. Giving priority to outgoing movements may thus generate much more busy scenarios within the system (e.g., more busy collectors, or not availability of FKL to move incoming items towards stacks), which are harder to face within the time limit imposed.

The SSL formulation seems to be much more effective in addressing the problem, being able to solve 131 out of 180 runs. Notice that it successfully solves all the instances when both NS-16 and NS-30 are coupled with the (10-10), (50-10) and (50-50) weight combinations, thus suggesting that formulation SSL, the time decomposition given by NS-16 and NS-30, and the weight combinations (10-10), (50-10) and (50-50) are appropriate settings for the efficiency of the proposed resolution approach. The option NS-10, instead, seems to be not suitable for generating subproblems that CPLEX can easily address within the given time limit. Moreover, as for SL, the weight combination $\psi = 10$ and $\xi = 50$ seems to generate too hard instances to tackle. This weight combination will be no longer discussed.

Table 3 reports the average solving times of the algorithm and some aggregated features of the solutions obtained by considering formulation SSL, NS-16 and NS-30, and weight combinations (10-10), (50-10) and (50-50), in terms of some crucial performance indicators suggested by our industrial partner, which are also used in stating the objective function (2) in terms of primary and secondary goals. Results are reported separately for NS-16 and NS-30. For each of these two options, the weight combinations (10-10), (50-10) and (50-50) correspond to the second, third and forth column in the table, respectively.

Table 3: Features of solutions (SSL formulation, options NS-16 and NS-30).

	NS-16		
(ψ - ξ)	(10 - 10)	(50 - 10)	(50 - 50)
Avg. Solving Time (min.)	28	25	40
LGV Avg. Travel Time (min.)	1,172	1,190	1,242
FKL Avg. Travel Time (min.)	1,783	1,918	1,862
Conveyors Avg. Idle Time per column (min.)	1.74	1.52	1.92
Collection area Saturation 100% (min.)	60	50	64
Collection area Saturation $\geq$ 90% (min.)	172	150	208
	NS-30		
(ψ - ξ)	(10 - 10)	(50 - 10)	(50 - 50)
Avg. Solving Time (min.)	8	7	11
LGV Avg. Travel Time (min.)	1,384	1,466	1,532
FKL Avg. Travel Time (min.)	2,024	2,120	2,184
Conveyors Avg. Idle Time per column (min.)	4.40	1.83	4.18
Collection area Saturation 100% (min.)	64	60	64
Collection area Saturation $\geq$ 90% (min.)	166	156	172

Specifically, Table 3 reports the average time in minutes the resolution approach needed to solve the 15 instances. Moreover, the primary goal of the industrial partner is analysed in terms of the average time in minutes travelled by all the 5 LGV and by all the 7 FKL, respectively, over the 15 instances (recall that each shift lasts 8 hours). Next, the secondary goals are addressed, i.e., the emptying of conveyors of incoming items and the anticipation movements of outgoing items toward the collection area. The first one is measured as the average time in minutes, over the 15 instances, incoming items idle on conveyors before been moved on an available collector, while the second one is measured in terms of the average saturation level of the collection area over the 15 instances. Regarding the saturation level of the collection area, two measures are reported here, i.e., the average time in minutes in which the collection area is completely saturated (Saturation 100%), and the average time in minutes in which the collection area is full at least at its 90% (Saturation $\geq$ 90%).

The version NS-30 seems to be faster in finding solutions with respect to NS-16, which however is still under the time limit imposed. Nevertheless, NS-30 is not able to optimize the travel time of the fleet of vehicles as good as NS-16 does, worsening the solutions of 18% for the travel time of the fleet of the LGV, and of 12% for the travel time of the fleet of the FKL with respect to NS-16 (on average over the three parameter settings). This may be explained by considering that increasing the number of subshifts surely defines smaller, and thus easier, problems to tackle, but at the same time may make the model myopic of the near future. This is confirmed also looking at the indicator Conveyors Avg. Idle Time, i.e., the average time of permanence of an incoming item on a conveyor. Results related to the exploitation of the collection area are quite similar for NS-16 and NS-30, with NS-30 slightly outperforming NS-16 in terms of the time the collection area is completely full. However, being the latter only a secondary goal of the company and coming at the expenses of a high increase of travel times for NS-30, NS-16 seems to address a more suitable time horizon splitting for the proposed resolution approach. Therefore, it is the only one discussed next. Regarding the weight combinations for NS-16, as expected, by increasing weight ψ from 10 to 50 and keeping $\xi = 10$, the idle time of incoming items on conveyors decreases, at the expense though of an increase of the average LGV and FKL travel times. Finally, the weight combination (10-10) outperforms the weight combination (50-50) in all the reported primary goals indicators. Moreover, by comparing the average solving times (reported in Table 3) of NS-16 with weight combinations (10-10) and (50-50), the latter appears to generate more tricky problems to tackle within the time limit imposed. Therefore, next only NS-16 with the weight combinations (10-10) and (50-10) is further discussed.

Table 4 and Table 5 report other features of the solutions obtained by considering formulation SSL, NS-16, and the weight combinations (10-10) and (50-10). Specifically, Table 4 shows the minimum, the maximum and the average time (in minutes) each vehicle has travelled over the 15 instances. Standard deviation is also reported. Table 5 shows instead the average time (in minutes) columns idle on the collection area before been loaded on trucks, the percentage of items being picked from their storage locations and directly moved to the collection area with no stop on collectors, the average time items spend idling on a collector separately for products in K_{in} and products in K_{out}, and finally the average time the collectors are full at least at their 60%. The latter is calculated as the average of the time in minutes all the 6 collectors are filled with a number of items exceeding the above mentioned saturation level over the 15 instances.

Table 4: Travel time details for the fleet of vehicles (in minutes).

	NS-16, $\psi = 10$, $\xi = 10$				NS-16, $\psi = 50$, $\xi = 10$			
LGV	Min.	Max.	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.
1	38	448	242	113.8	42	400	236	101.7
2	42	416	232	113.3	40	434	235	107.6
3	58	418	233	108.7	54	424	233	105.0
4	50	430	234	107.4	48	422	244	104.0
5	58	426	230	106.5	54	410	242	95.7
FKL	Min.	Max.	Avg.	Std. Dev.	Min.	Max.	Avg.	Std. Dev.
1	40	474	265	128.5	34	476	282	131.0
2	32	476	248	132.6	42	476	265	134.4
3	0	480	248	139.9	0	476	271	136.9
4	34	480	247	141.6	40	468	272	127.2
5	26	480	257	135.9	24	480	271	137.9
6	30	480	255	135.4	28	470	280	129.7
7	36	470	261	133.0	26	480	277	127.7

Table 5: collection area and collectors details.

(ψ - ξ)	NS-16 (10 - 10)	NS-16 (50 - 10)
Avg. Idle Time in Collection area per column (min.)	386	357
Qty directly to Collection area	92%	93%
Avg. Idle Time on Collectors per column K_{in} (min.)	1.50	1.20
Avg. Idle Time on Collectors per column K_{out} (min.)	7.2	5.1
Saturation of Collectors $\geq 60\%$ (min.)	20	20

As outlined in Table 4, travel times for the same type of vehicle seem to be quite balanced on average for both types of weight combinations. Moreover, according to Table 5, prioritizing the emptying of conveyors, i.e., increasing ψ from 10 to 50, not only causes a decrease of idle time of incoming items on conveyors, as observed before, but also a decrease of idle time of incoming items on collectors. Incoming items are thus faster moved from the conveyors towards their assigned stacks when the weight combination (50-10) is chosen.

Prioritizing the conveyors emptying movements does also affect the movements of outgoing items. In fact, the number of outgoing items being retrieved from their stacks and directly transported to the collection area is slightly increased, implying a lower exploitation of collectors by outgoing items as well as a decrease of the average time outgoing items spend idling on collectors. Also observe that the average idle time of outgoing items in the collection area decreases of about the 7% when the weight combination (50-10) is chosen. This may be explained by considering that, when using the weight combination (50-10), the movements towards the collection area are delayed in order to prioritize the movements of incoming freight from collectors to stacks performed by FKL. Outgoing items are thus retrieved from stacks later than when the weight combination (10-10) is chosen, idling less time in the collection area.

Finally, note that the average idle time of incoming and outgoing items on collectors is very low when considering both weight combinations, and that the saturation of collectors exceeds the 60% of their capacities for only a few minutes on average, thus testifying a very good synchronization among vehicles for the movements of items, so avoiding congestion on collectors.

By summarizing, decomposing each shift into 16 subshifts of equal length, and solving the resulting subproblems via formulation SSL, under either the setting (10-10) or the setting (50-10) for parameters ψ and ξ, appears to be an efficient algorithmic strategy to solve the addressed SRP, by obtaining solutions of good quality in terms of travel times of the vehicles and their synchronization, and also in terms of an effective exploitation of collectors and collection area within the warehouse.

6.5 Worst-case scenario analysis

For the worst-case scenario analysis, we have considered one of the busiest weeks for the company with respect to both production and shipments, just before a peak period of requests. Indeed, in the selected week both production and shipments are higher of about the 25% with respect to a normal week, and about 500 more movements are required for storing or retrieving items per shift. Days are solved in cascade, from the first shift of the first day of the week till the last one.

We considered formulation SSL, and we used the option NS-16, i.e., we split each shift into 16 subshifts, and the weight combination $\psi = 10$ and $\xi = 10$. The main motivation for considering the weight combination (10-10) is that, working on a weekly basis and focusing on a week with a very high rotation index, both storing and retrieving operations appear to be particularly crucial to manage, and therefore any sort of prioritization might bring to too expensive results in terms of algorithm solving time.

Under the considered setting, the matheuristic approach we propose is able to determine a solution to all the shifts composing the week under study. Table 6 summarizes the same kinds of results reported in Table 3 and Table 5. In particular, the first column refers to the busy week under study, the second column summarizes the results already reported in Table 3 and Table 5 for option NS-16 and the weight combination $\psi = 10$ and $\xi = 10$, which refer to an ordinary number of operations within the warehouse, and the third column shows the difference in percentage between the the first two columns.

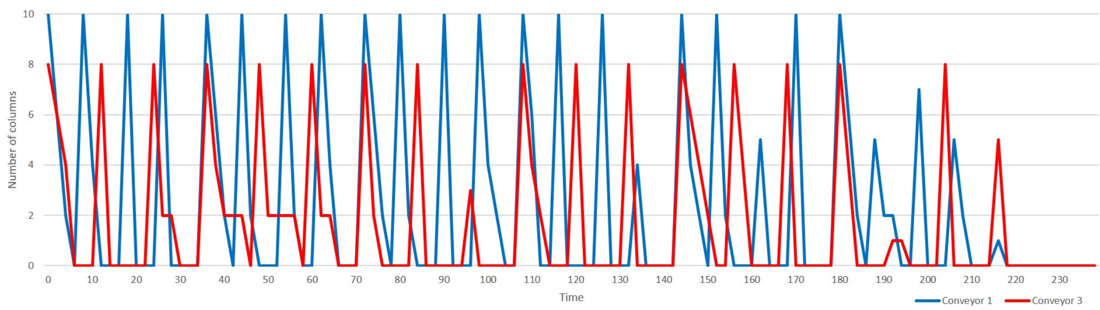
Table 6: Features of solutions for NS-16 in a worst-case scenario.

NS-16, $\psi = 10$, $\xi = 10$			
	Busy	Ordinary	% Diff.
Avg. Solving Time (min.)	43	28	+54%
LGV Avg. Travel Time (min.)	1,379	1,172	+18%
FKL Avg. Travel Time (min.)	2,167	1,783	+22%
Conveyor Avg. Idle Time (min.)	1.67	1.74	−4%
Collection area Saturation 100% (min.)	46	60	−23%
Collection area Saturation ≥ 90% (min.)	144	172	−16%
Avg. Idle Time in Collection area (min.)	399	386	+3%
Qty directly to Collection area	96%	92%	+4%
Avg. Idle Time on Collectors per column K_{in} (min.)	1.1	1.5	−27%
Avg. Idle Time on Collectors per column K_{out} (min.)	9.4	7.2	+31%
Saturation of Collectors ≥ 60% (min.)	18	20	−10%

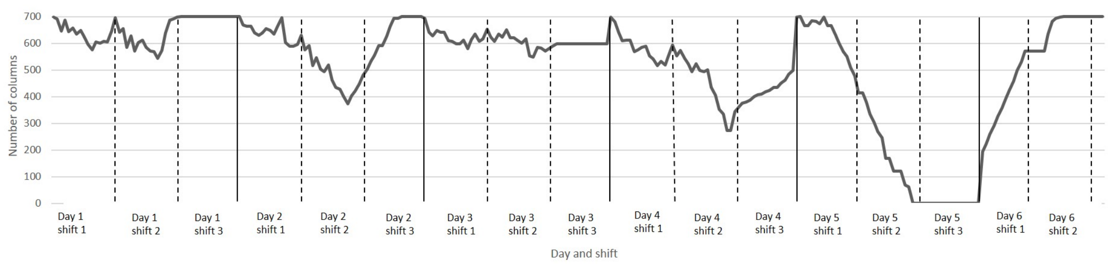
The increased number of movements requested in the selected busy week causes an unavoidable increase of travel times for both LGV and FKL (+18% and +22%, respectively). Conveyors are strongly used in this busy week and, being releases more frequent than in ordinary periods, they are required to be emptied by LGV in a faster way not to block the production of the site (recall that production decisions are independent of warehouse management). Indeed, the idle time on conveyors of incoming items is slightly decreased with respect to the more ordinary shifts (of about the 4%). Similarly, the average idle time of incoming items on collectors is decreased (of about the 27%). Therefore, faster movements of incoming items from conveyors to stacks are performed in this busy week with respect to more ordinary weeks.

Direct movements of outgoing items from stacks towards the collection area are increased in this busy week (compare the indicator "Qty directly to Collection area"). Nevertheless, for those outgoing items passing through a collector on their itinerary towards the collection area, a longer (+31%) idle time on collectors is observable. Additionally, the collection area is saturated for less time (compare the indicators “Collection area Saturation 100% and 90%” for both scenarios), but anticipation movements are performed with large advance, as testified by the longer (+3%) idle time of columns in the collection area.

Figure 2a and Figure 2b report the saturation trends of two crucial spots of the warehouse for specific periods. Figure 2a shows the number of columns released and idling on conveyor 1



(a) Saturation level of conveyors 1 and 3 in shift 1 of day 1.



(b) Saturation of the collection area.

Figure 2: Saturation trends of conveyors and collection area.

and conveyor 3 during the first shift of day 1 of the considered week. Notice that the statistics for conveyor 2 is not reported as the amount of released items is almost 0, thus conveyors 1 and 3 are extremely exploited. The selected day has both a production rate and a shipment request higher than the average calculated over all the shifts of the week. The unit of measure of the time reported on the x-axis is 4 minutes. The capacity of conveyor 1 and conveyor 3 is 10 and 8 columns, respectively. The LGV empty conveyors with large advance with respect to new releases, thus avoiding production delays caused by busy conveyors. Only a small amount of items remains idling for a long time, which is however less than 30 minutes. Figure 2b reports instead the number of columns idling in the collection area during the entire week (the last shift of the last day is not reported as the saturation has been already reached during the previous shift). In general, the collection area is well exploited. Especially during the third shift of each day (i.e., the night shift), a very high number of columns are moved towards the collection area. Recall that, during this shift, production still continues and storage operations are also required, so workers are not dedicated to replenishment only. This behaviour is clearly showed during the third shift of day 2 and day 4. At the end of day 5, the collection area is completely emptied. This is because no shipments are planned on day 6. However, on day 6 the shipping list of the first day of the next week is available and replenishment of the collection area can start again. Saturation is reached during the second shift of that day.

Finally, regarding the average solving time required by the approach to solve a shift of the worst-case week under consideration, it is much lower (see Table 6) than the time the company requires to solve a shift, that is 4 hours. The proposed matheuristic appears thus to be a valuable tool for solving the considered SRP problem also in real worst-case scenarios.

7 Conclusions

This paper discusses a sequencing and routing problem originated from a real-world application context in tissue logistics. Specifically, the problem consists in defining the best sequence of locations to visit within a warehouse for the storage and/or retrieval of a given set of items during a specified time horizon, by considering some additional requirements. In particular, an anticipation movements policy and a strict order of precedence to fill and retrieve items in/from storage locations have to be considered when planning the operations of two fleets of different types of vehicles, having movements restrictions within the warehouse. The first policy is pursued due to the high number of movements daily requested, with the scope of anticipating operations with respect to peak and very busy periods. On the other hand, the order of precedence is pursued due to the perishability of the products managed within the warehouse.

We have modelled the problem as a constrained multicommodity flow problem on an space-time network, and we have proposed a Mixed-Integer Linear Programming formulation, with some enhancements, as well as a matheuristic approach based on the decomposition of the time horizon. Precisely, the original problem has been split in subproblems that can be easily addressed via a state-of-the-art optimization solver, and solved in cascade. A wide experimental analysis has been then presented by considering real instances provided by our industrial partner. The reported computational results show the efficiency and the efficacy of the proposed approach. We plan to extend the achieved results by studying a combined optimization problem which integrates the pick up and put away operations with assignment storage location decisions. The assignment of storage locations, the scheduling of put away and pick up operations, and the routing of the vehicles inside the warehouse define in fact hard interdependent decisions which are very challenging to address.

Appendix A

This appendix is devoted to specify the set of nodes and arcs composing the subgraphs $\mathcal{G}_{in} = (\mathcal{N}_{in}, \mathcal{A}_{in})$, $\mathcal{G}_{out} = (\mathcal{N}_{out}, \mathcal{A}_{out})$, $\mathcal{G}_{F1} = (\mathcal{N}_{F1}, \mathcal{A}_{F1})$ and $\mathcal{G}_{F2} = (\mathcal{N}_{F2}, \mathcal{A}_{F2})$, defined in Section 4, where commodities $k \in \mathcal{K}_{in}$ and $k \in \mathcal{K}_{out}$, and vehicles $v \in \mathcal{V}^1$ and $v \in \mathcal{V}^2$ may move, respectively.

Incoming freight flow of product type $k \in \mathcal{K}_{in}$ is originated from an input point $r \in \mathcal{R}$, it passes through some collectors $b \in \mathcal{B}$, it may possibly pass through some storage locations in $\mathcal{S}_{out}$, and some storage locations $s^{k'} \in \mathcal{S}_{in}^{k'}$, with $k' \in \mathcal{K}_{in} \setminus \{k\}$, finally reaching its assigned storage location $s^k \in \mathcal{S}_{in}^k$. Thus, the set of nodes $\mathcal{N}_{in}$ is defined as follows:

$$\mathcal{N}_{in} := \{(i,t) \in \mathcal{N} : \ i \in \mathcal{R} \cup \mathcal{B} \cup \mathcal{S}_{out} \cup \mathcal{S}_{in}\}.$$

The set of movement arcs for incoming freight flow of product type $k \in \mathcal{K}_{in}$ is composed of arcs defined as follows:

- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{R},\ j \in \mathcal{B}$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{R},\ j \in \mathcal{R},\ i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{B},\ j \in \mathcal{S}_{out} \cup \mathcal{S}_{in}$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{B},\ j \in \mathcal{B},\ i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\ j \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\ i \neq j$.

The set of holding arcs for incoming freight flow of product type $k \in \mathcal{K}_{in}$ is composed of arcs defined as follows:

$$\{((i,t),(i,t+1)) \in \mathcal{A}:\ (i,t) \in \mathcal{N}_{in},\ (i,t+1) \in \mathcal{N}_{in}\}.$$

Outgoing freight flow of product type $k \in \mathcal{K}_{out}$ is originated from a storage location $s^k \in \mathcal{S}^k_{out}$, it may possibly pass through some storage locations $s^{k'} \in \mathcal{S}^{k'}_{out}$, with $k' \in \mathcal{K}_{out} \setminus \{k\}$, and some storage locations in $\mathcal{S}_{in}$, the collectors $b \in \mathcal{B}$, finally reaching the collection area π. Thus, the set of nodes $\mathcal{N}_{out}$ is defined as follows:

$$\mathcal{N}_{out} := \{(i,t) \in \mathcal{N}:\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in} \cup \mathcal{B} \cup \{\pi\}\}.$$

The set of movement arcs for outgoing freight flow of product type $k \in \mathcal{K}_{out}$ is composed of arcs defined as follows:

- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\ j = \pi$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{S}_{out} \cup S_{in},\ j \in \mathcal{B}$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\ j \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\ i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{B},\ j = \pi$;
- $((i,t),(j,t')) \in \mathcal{A}:\ i \in \mathcal{B},\ j \in \mathcal{B},\ i \neq j$.

The set of holding arcs for outgoing freight flow of product type $k \in \mathcal{K}_{out}$ is composed of arcs defined as follows:

$$\{((i,t),(i,t+1)) \in \mathcal{A}:\ (i,t) \in \mathcal{N}_{out},\ (i,t+1) \in \mathcal{N}_{out}\}.$$

A vehicle of type $v \in \mathcal{V}^1$ may only move in the hallway, between the input points $r \in \mathcal{R}$, the collectors $b \in \mathcal{B}$ and the parking area ω^1. Thus, the set of nodes $\mathcal{N}_{F1}$ is defined as follows:

$$\mathcal{N}_{F1} := \left\{(i,t) \in \mathcal{N}:\ i \in \mathcal{R} \cup \mathcal{B} \cup \left\{\omega^1\right\}\right\}.$$

The set of movement arcs of vehicle $v \in \mathcal{V}^1$ is composed of arcs defined as follows:

- $((i,t),(j,t')) \in \mathcal{A} : i = \omega^1,\ j \in \mathcal{R}$;
- $((i,t),(j,t')) \in \mathcal{A} : i \in \mathcal{R},\ j \in \mathcal{B}$;
- $((i,t),(j,t')) \in \mathcal{A} : i \in \mathcal{R},\ j \in \mathcal{R},\ i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A} : i \in \mathcal{B},\ j \in \mathcal{R}$;

- $((i,t),(j,t')) \in \mathcal{A} : i \in \mathcal{B},\, j = \omega^1$;
- $((i,t),(j,t')) \in \mathcal{A} : i \in \mathcal{B},\, j \in \mathcal{B},\, i \neq j$.

The set of holding arcs of vehicle $v \in \mathcal{V}^1$ is composed of arcs defined as follows:

$$\{((i,t),(i,t+1)) \in \mathcal{A} :\ (i,t) \in \mathcal{N}_{F1},\ (i,t+1) \in \mathcal{N}_{F1}\}.$$

A vehicle of type $v \in \mathcal{V}^2$ may only move in the storage area, between the collectors $b \in \mathcal{B}$, the sets of storage locations $\mathcal{S}_{out}$ and $\mathcal{S}_{in}$, the collection area π and the parking area ω^2. Thus, the set of nodes $\mathcal{N}_{F2}$ is defined as follows:

$$\mathcal{N}_{F2} := \left\{(i,t) \in \mathcal{N} :\ i \in \mathcal{B} \cup \mathcal{S}_{out} \cup \mathcal{S}_{in} \cup \{\pi\} \cup \left\{\omega^2\right\}\right\}.$$

The set of movement arcs of vehicle $v \in \mathcal{V}^2$ is composed of arcs defined as follows:

- $((i,t),(j,t')) \in \mathcal{A} :\ i = \omega^2,\, j \in \mathcal{S}_{out}$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i = \omega^2,\, j \in \mathcal{B}$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{S}_{in} \cup \mathcal{B},\, j = \omega^2$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{S}_{out} \cup \mathcal{B},\, j = \pi$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{B},\, j \in \mathcal{S}_{out} \cup \mathcal{S}_{in}$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{B},\, j \in \mathcal{B},\, i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\, j \in \mathcal{B}$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\, j \in \mathcal{S}_{out} \cup \mathcal{S}_{in},\, i \neq j$;
- $((i,t),(j,t')) \in \mathcal{A} :\ i = \pi,\, j = \omega^2$.

The set of holding arcs of vehicle $v \in \mathcal{V}^2$ is composed of arcs defined as follows:

$$\{((i,t),(i,t+1)) \in \mathcal{A} :\ (i,t) \in \mathcal{N}_{F2},\ (i,t+1) \in \mathcal{N}_{F2}\}.$$

Acknowledgements

The research has been supported by Region of Tuscany-Regional Government (POR FESR 2014-2020-Line 1-Research and Development Strategic Projects) through the Project IREAD4.0 under Grant CUP 7165.24052017.112000028.

References

Ballestín, F., Pérez, Á., Lino, P., Quintanilla, S., and Valls, V. (2013). Static and dynamic policies with rfid for the scheduling of retrieval and storage warehouse operations. *Computers & Industrial Engineering*, 66(4):696–709.

Ballestín, F., Pérez, Á., and Quintanilla, S. (2020). A multistage heuristic for storage and retrieval problems in a warehouse with random storage. *International Transactions in Operational Research*, 27(3):1699–1728.

Boysen, N., Briskorn, D., and Emde, S. (2017). Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research*, 259(1):293–307.

Briant, O., Cambazard, H., Cattaruzza, D., Catusse, N., Ladier, A.-L., and Ogier, M. (2020). An efficient and general approach for the joint order batching and picker routing problem. *European Journal of Operational Research*.

Carlo, H. J., Vis, I. F., and Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2):412–430.

Chabot, T., Lahyani, R., Coelho, L. C., and Renaud, J. (2017). Order picking problems under weight, fragility and category constraints. *International Journal of Production Research*, 55(21):6361–6379.

Chen, F., Wang, H., Qi, C., and Xie, Y. (2013). An ant colony optimization routing algorithm for two order pickers with congestion consideration. *Computers & Industrial Engineering*, 66(1):77–85.

Chen, F., Wang, H., Xie, Y., and Qi, C. (2016). An aco-based online routing method for multiple order pickers with congestion consideration in warehouse. *Journal of Intelligent Manufacturing*, 27(2):389–408.

Cinar, D., Oliveira, J. A., Topcu, Y. I., and Pardalos, P. M. (2017). Scheduling the truckload operations in automated warehouses with alternative aisles for pallets. *Applied Soft Computing*, 52:566–574.

Cortés, P., Gómez-Montoya, R. A., Muñuzuri, J., and Correa-Espinal, A. (2017). A tabu search approach to solving the picking routing problem for large-and medium-size distribution centres considering the availability of inventory and k heterogeneous material handling equipment. *Applied Soft Computing*, 53:61–73.

Crainic, T. G., Ricciardi, N., and Storchi, G. (2009). Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4):432–454.

Cuda, R., Guastaroba, G., and Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199.

Davarzani, H. and Norrman, A. (2015). Toward a relevant agenda for warehousing research: literature review and practitioners' input. *Logistics Research*, 8(1):1.

de Brito, M. P. and De Koster, M. (2004). Product and material returns: handling and warehousing issues. In *Reverse Logistics*, pages 135–153. Springer.

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

De Santis, R., Montanari, R., Vignali, G., and Bottani, E. (2018). An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses. *European Journal of Operational Research*, 267(1):120–137.

Gagliardi, J.-P., Renaud, J., and Ruiz, A. (2012). Models for automated storage and retrieval systems: a literature review. *International Journal of Production Research*, 50(24):7110–7125.

Gómez-Montoya, R. A., Cano, J. A., Cortés, P., and Salazar, F. (2020). A discrete particle swarm optimization to solve the put-away routing problem in distribution centres. *Computation*, 8(4):99.

Gong, Y. and De Koster, R. B. (2011). A review on stochastic models and analysis of warehouse operations. *Logistics Research*, 3(4):191–205.

Grosse, E. H., Glock, C. H., Jaber, M. Y., and Neumann, W. P. (2015). Incorporating human factors in order picking planning models: framework and research opportunities. *International Journal of Production Research*, 53(3):695–717.

Gu, J., Goetschalckx, M., and McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21.

Gue, K. R., Ivanović, G., and Meller, R. D. (2012). A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review*, 48(4):795–806.

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228.

Kubo, M. and Kasugai, H. (1991). The precedence constrained traveling salesman problem. *Journal of the Operations Research Society of Japan*, 34(2):152–172.

Küçükyaşar, M., Y. Ekren, B., and Lerher, T. (2020). Cost and performance comparison for tier-captive and tier-to-tier sbs/rs warehouse configurations. *International Transactions in Operational Research*.

Lanza, G., Passacantando, M., and Scutellà, M. G. (2021). *Assigning and sequencing storage locations under a two level storage policy: optimization model and matheuristic approach*. Technical Report 2398. University of Pisa.

Lu, W., McFarlane, D., Giannikas, V., and Zhang, Q. (2016). An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research*, 248(1):107–122.

Masae, M., Glock, C. H., and Grosse, E. H. (2020). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, 224:107564.

Matusiak, M., de Koster, R., Kroon, L., and Saarinen, J. (2014). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, 236(3):968–977.

Mowrey, C. H. and Parikh, P. J. (2014). Mixed-width aisle configurations for order picking in distribution centers. *European Journal of Operational Research*, 232(1):87–97.

Oliveira, J. A. (2007). Scheduling the truckload operations in automatic warehouses. *European Journal of Operational Research*, 179(3):723–735.

Pan, J. C.-H. and Wu, M.-H. (2012). Throughput analysis for order picking system with multiple pickers and aisle congestion considerations. *Computers & Operations Research*, 39(7):1661–1672.

Pohl, L. M., Meller, R. D., and Gue, K. R. (2009a). An analysis of dual-command operations in common warehouse designs. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):367–379.

Pohl, L. M., Meller, R. D., and Gue, K. R. (2009b). Optimizing fishbone aisles for dual-command operations in a warehouse. *Naval Research Logistics*, 56(5):389–403.

Ratliff, H. D. and Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, 31(3):507–521.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G.-J., Mantel, R., and Zijm, W. H. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, 122(3):515–533.

Scholz, A., Henn, S., Stuhlmann, M., and Wäscher, G. (2016). A new mathematical programming formulation for the single-picker routing problem. *European Journal of Operational Research*, 253(1):68–84.

Schrotenboer, A. H., Wruck, S., Roodbergen, K. J., Veenstra, M., and Dijkstra, A. S. (2017). Order picker routing with product returns and interaction delays. *International Journal of Production Research*, 55(21):6394–6406.

Tappia, E., Roy, D., Melacini, M., and De Koster, R. (2019). Integrated storage-order picking systems: Technology, performance models, and design insights. *European Journal of Operational Research*, 274(3):947–965.

van Gils, T., Ramaekers, K., Caris, A., and de Koster, R. B. (2018). 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.

Wang, Y., Liu, Z., Huang, K., Mou, S., and Zhang, R. (2020). Model and solution approaches for retrieval operations in a multi-tier shuttle warehouse system. *Computers & Industrial Engineering*, 141:106283.

Weidinger, F., Boysen, N., and Schneider, M. (2019). Picker routing in the mixed-shelves warehouses of e-commerce retailers. *European Journal of Operational Research*, 274(2):501–515.

Wruck, S., Vis, I. F., and Boter, J. (2013). Time-restricted batching models and solution approaches for integrated forward and return product flow handling in warehouses. *Journal of the Operational Research Society*, 64(10):1505–1516.

Zhao, X., Zhang, R., Zhang, N., Wang, Y., Jin, M., and Mou, S. (2020). Analysis of the shuttle-based storage and retrieval system. *IEEE Access*, 8:146154–146165.

Žulj, I., Glock, C. H., Grosse, E. H., and Schneider, M. (2018). Picker routing and storage-assignment strategies for precedence-constrained order picking. *Computers & Industrial Engineering*, 123:338–347.