Optimization of a fleet of reconfigurable robots

Mari Chaikovskaia[a,*], Jean-Philippe Gayon[a], Alain Quilliot[a]

[a] *Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont Auvergne INP, LIMOS, 63000 Clermont-Ferrand, France*

Abstract

We consider a fleet of elementary robots that can be connected in different ways to transport loads of different types. For instance, a single robot can transport a small load and the association of two robots can either transport a large load or two small loads. The robot associations can be reconfigured between two trips. We seek to determine the minimum number of robots necessary to transport a set of loads in a given time interval. We formulate this fleet sizing problem with an integer linear program. We also derive analytical expressions for the minimum number of robots in the special case of unit capacities. Finally, we compare the minimum number of robots with or without reconfiguration. We show that the value of reconfigurability can be very high and diminishes with the fleet size. Reconfigurability is particularly useful when the demand for small loads has to be met at a different time interval from the demand for large loads. Finally, numerical experiments show that we can obtain the optimal solution in a short computation time when the number of load types and configurations is reasonable, which corresponds to many warehouse configurations.

Keywords : Logistics; Fleet sizing; Reconfigurability; Robots; Warehouse.

1. Introduction

This research is the fruit of a collaboration between LIMOS CNRS Laboratory and MecaBotiX company. This company designs reconfigurable mobile poly-robots that transport standardized loads in a warehouse such that boxes or pallets. A poly-robot is a vehicle formed by assembling multiple elementary robots (or simply bots). These poly-robots can be reconfigured over time to adapt to the type of load to be carried. Figure 1 shows four examples of M3-Cooper poly-robots: a mono-bot, a bi-bot, a tri-bot and a quadri-bot. For instance, a box could be carried by a mono-bot, while a pallet could be carried by a quadri-bot.

*Corresponding author : j-philippe.gayon@uca.fr

Email addresses: `mari.chaikovskaia@doctorant.uca.fr` (Mari Chaikovskaia), `j-philippe.gayon@uca.fr` (Jean-Philippe Gayon), `alain.quilliot@uca.fr` (Alain Quilliot)

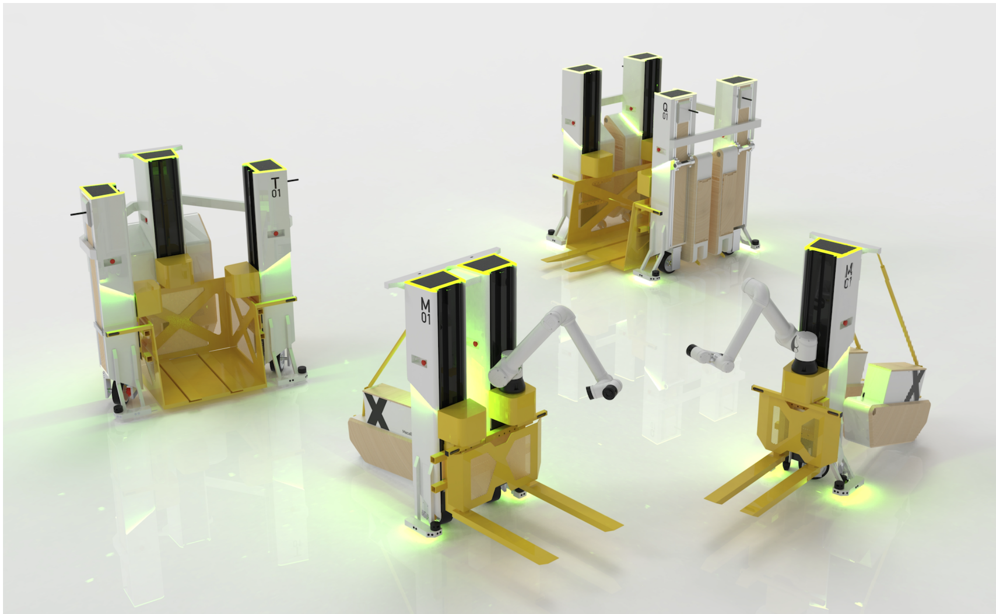


Figure 1: Examples of M3-Cooper reconfigurable poly-robots (MecaBotiX, 2024)

Reconfigurable poly-robots present promising advantages for future warehouse transport operations. First, they can dynamically adjust to the size or weight of the load, eliminating the need for oversized or undersized robots and enabling the reassignment of available modular units. Second, these robots can facilitate denser storage by navigating narrower aisles when operating as single units. Lastly, their interchangeable design enhances fault tolerance and lowers maintenance expenses.

However, a number of challenges remain before reconfigurable robots can be used on an industrial scale in logistics warehouses. First of all, there are technical issues to be resolved in order to coordinate the robots with each other and enable rapid, automated reconfiguration. In addition, there are several optimization issues, such as task scheduling and fleet sizing. In this paper, we focus on the problem of sizing a fleet of reconfigurable robots. We consider two scenarios. In the first, reconfiguration is only possible at the beginning and configurations are fixed for the entire time horizon. In the second, the poly-robots can be reconfigured over time. The objective is then to determine and compare the minimum number of bots required to move a set of loads within a given time horizon, for both scenarios.

2. Literature review

In this section, we review the literature related to reconfigurable robotic systems and fleet sizing problem. We also provide a summary of our contributions.

Reconfigurable robotic systems

A reconfigurable robotic system is an assembly of modules that can attach and detach from each other to modify and adapt to different tasks and environments (Bojinov et al., 2000). Stoy et al. (2010) define three categories of self-reconfigurable robots based on the number of modules: *pack* robots, *herd* robots and *swarm* robots. Pack robots are composed of several modules, usually in the range of tens, and requires strict coordination due to the fact that the individual modules play a crucial role in the robot's overall performance. Herd robots are composed of a large number of modules, usually in the range of hundreds, and global coordination of these modules is challenging. They are better managed as a collection of groups since the actions of individual modules are still significant but not as critical to the overall performance of the robot. Eventually, swarm robots are comprised of countless modules. Here, each module is controlled locally since the impact of an individual module on the overall behavior of the robot is minimal.

Several prototypes of reconfigurable robotic systems have been developed. The reader is referred to Jahanshahi et al. (2017) or Seo et al. (2019) for a survey on this topic. For example, systems using multiple modules can create different forms to perform different tasks: it could turn into a snake to reach narrow places, into a hexapod to carry a load or it may split into many smaller robots to perform a task in parallel (Castano et al., 2000; Yim et al., 2000). The self-reconfigurable robots can also be used as conveyors. The spherical shape of the ATRON modules enables them to function as wheels, facilitating the construction of surfaces that have the ability to transport items (Østergaard et al., 2006; Brandt et al., 2007). Shen et al. (2006) demonstrate a solution based on SuperBot modules that can perform multimodal locomotions such as snake, caterpillar, insect, spider, rolling track, H-walker, etc. Chebab (2018) focuses on the design of new architectures of modular mobile manipulators that can cooperate with each other to perform tasks in industrial or service contexts related to the handling and transport of boxes. Wan et al. (2024) propose a navigation framework with non-complex transformation states for inter-reconfigurable robots to perform combining and splitting control dimensions. Another application of reconfigurable robots can be found in Mars exploration, where tasks such as transportation or building construction have to be performed with limited resources (Irawan et al., 2019). A survey of modular system for multifunctional applications in space exploration is presented by Post et al. (2021). More generally, reconfigurable robotic systems are related to reconfigurable manufacturing systems (see e.g. Cui et al. (2024)).

Fleet sizing problem

The fleet sizing problem consists in determining the optimal number of vehicles for the transport of goods, this is a key logistics problem which concerns all means of transport (air, sea, road, inside warehouses, ...). Baykasoğlu et al. (2019) provides a review of fleet planning problems (including fleet sizing) in transportation systems. In road transportation, the issue is not only the composition of the fleet but also the choice of the route (Hoff et al., 2010). The problem involves tanks and rail cars (Sha and Srinivasan, 2016; Milenković and Bojović, 2013; Cheon et al., 2012), trucks (Mohtasham et al., 2021; Amjath et al., 2022), vehicles (Rahimi-Vahed et al., 2015; Koç et al., 2014; Kumar et al., 2018) and electric vehicles (Miao et al., 2020; Manzolli et al., 2022; Guo et al., 2024a,b). With respect to maritime fleets, several studies consider a single type of vessel (Everett et al., 1972; Lai and Lo, 2004), while others extend the approach to the case of several types of vessels (Schwartz, 1968; Mehrez et al., 1995). The problem of renewing the maritime fleet consists in dynamically adjusting the fleet according to the evolution of the service requirements (Xinlian et al., 2000; Meng and Wang, 2011). A complete review on the maritime fleet size problem is given by Pantuso et al. (2014).

Finally, we mention here a study that considers vehicles with configurable capacity. Tellez et al. (2018) consider several types of users with different spatial requirements such as passengers using seats or wheelchairs. They explore a variant of the dial-a-ride problem that allows for on-route modifications to the vehicle's interior configuration.

The fleet sizing problem has also been investigated in the field of autonomous vehicles which include Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs). Recent review paper of Leong and Ahmad (2024) gives a detailed overview of the autonomous load-carrying mobile robots, with a particular focus on indoor applications for both ground and aerial platforms. Vis (2006) reviews research on AGV design and control and Fragapane et al. (2021) examines AMR planning and control for intralogistics. Vis (2006) identifies three categories of fleet sizing models: deterministic, stochastic and simulation. Deterministic approaches like network flow models and linear programming models can be utilized prior to the actual operation to estimate the required number of vehicles. Stochastic models, such as queuing networks or simulation models, aim to incorporate external influences (see e.g. Koo et al. (2004); Choobineh et al. (2012); Gödeke and Detzner (2023); Soufi et al. (2024)). On one hand, analytical models tend to underestimate the required number of vehicles compared to simulation results. On the other hand, simulation requires a lot of details and hardly copes with large fleets.

In what follows, we provide details on deterministic approaches for sizing autonomous vehicle fleets. Lee and Murray (2019) investigate a new approach for warehouse order picking. They focus on two types of commercially available mobile robots: pickers, capable of grasping items from shelves, and transporters, designed to swiftly deliver items from the warehouse to the packing station. They determine the optimal combination of picker and transport robots that surpasses the performance of traditional human-based picking operations. Lyu et al. (2019) simultaneously consider the optimal number of AGVs, the shortest transportation time, a path planning problem, and a conflict-free routing problem. Aziez et al. (2021) focus on the optimization of the number and types of carts and AGVs required to fulfill daily requests in a hospital while optimizing AGVs routes and adhering to time constraints. Each request necessitates specific types of carts, which are transported by the AGVs.

Contributions

We consider the problem of sizing a fleet of cooperative and reconfigurable robots to transport loads of different types. Our main contributions can be summarized as follows. In Section 3, we propose a framework to model the fleet sizing problem for a fleet of reconfigurable robots. In Section 4, we propose Integer Linear Programs (ILPs) formulations for the problems with or without reconfiguration. In Section 5, we derive closed-form expressions for the minimum number of robots in the special case of unit capacities. In Section 6, we compare the strategy with reconfiguration and the strategy without reconfiguration. We show that reconfiguration can allow to reduce significantly the fleet size. However, the value of reconfigurability diminishes with the fleet size. We also consider a variant where the demand is per period rather than over the whole horizon. For this variant, we show that the gain, in number of robots, is not bounded. Finally, in Section 7 we present numerical experiments that show that our ILP models can be solved in a very short time for small to medium instances.

3. Assumptions and notations

We consider a fleet of N mobile elementary robots that are able to cooperate to transport loads of different types. An elementary robot is abbreviated to bot. A *poly-robot* refers to a vehicle obtained by combining together several bots. A p-bot refers to a poly-robot involving exactly p bots. We call the number p a *configuration*. A 1-bot is a bot working alone. A maximum of P bots can cooperate together ($p = 1, \cdots, P$).

There are d_k loads of type k to be transported ($k = 1, \cdots, K$). We assume that the demand is known at the beginning of the time horizon. All the loads to be moved are located in the loading area of the warehouse and must be transported to the unloading area. A p-bot can only carry one load type at a time and can simultaneously carry c_{pk} loads of type k. We assume that for each load type there is at least one configuration capable of carrying it.

The time horizon is divided into T periods ($t = 1, \cdots, T$). During a period, a p-bot is able to perform a round trip and carry a maximum of c_{pk} loads of type k. At each period, the poly-robots can reconfigure themselves. For example, if we have at time t a 5-bot and a 2-bot, that is to say a total of 7 bots, we can transform them into a 3-bot and a 4-bot in period $t + 1$ (see Figure 2).

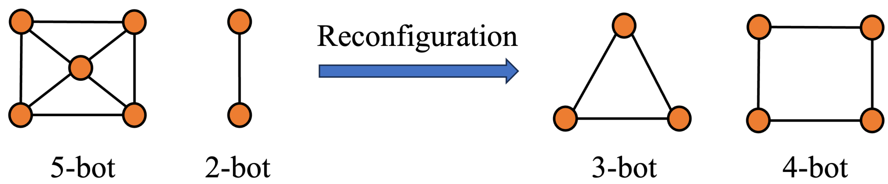


Figure 2: Reconfiguration example between two periods

The goal is to minimize the number of bots needed to carry all the loads over the time horizon. We consider two variants. In the first variant, reconfiguration is prohibited and the configurations are decided for the entire horizon. In the second variant, reconfiguration is allowed and the configurations can be changed at the start of each period. When reconfiguration is allowed, we denote by N_R the minimum number of bots. When reconfiguration is not allowed, we denote by N_W the minimum number of bots.

We summarize below the main notations that will be used in subsequent sections.

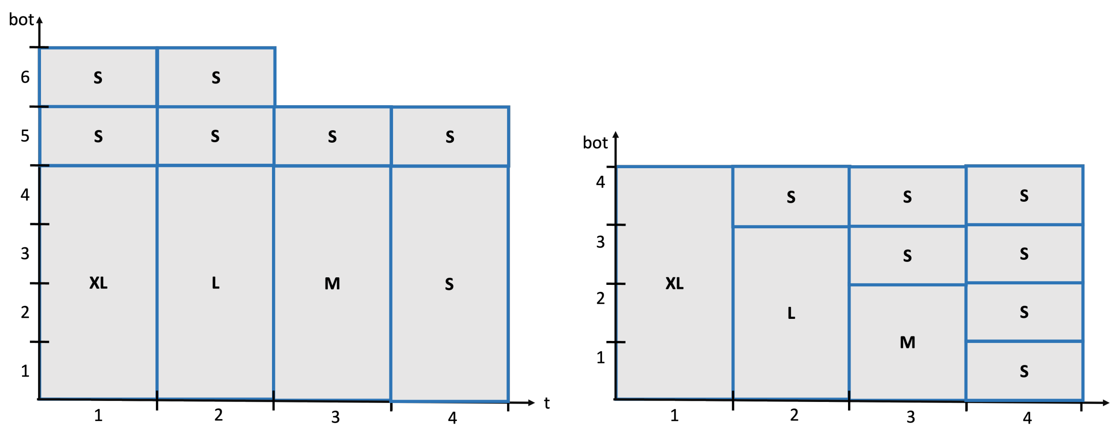
P	Number of configurations ($p = 1, \cdots, P$)
p-bot	Poly-robot formed with p bots
K	Number of load types ($k = 1, \cdots, K$)
d_k	Number of loads of type k
c_{pk}	Number of loads of type k that a p-bot can transport in a single period
T	Number of periods ($t = 1, \cdots, T$)
N_W	Minimum number of bots when reconfiguration is forbidden
N_R	Minimum number of bots when reconfiguration is allowed
x_{pk}^t	Number of p-bots transporting loads of type k in period t
x_p	Number of p-bots (only defined for the problem without configuration)

Example. We now present an example to illustrate the problems with or without configuration. Consider four configurations, as in Figure 1, and four types of loads 1, 2, 3, 4 corresponding to respectively Small (S), Medium (M), Large (L) and Extra Large (XL) loads. Let's take $T = 5$, $d_1 = 7$, $d_2 = 1$, $d_3 = 1$ and $d_4 = 1$. The capacities are set to 0 or 1 and are given by Table 1.

The optimal solutions are trivial for this simple problem and are represented with a Gantt chart in Figure 3. Each rectangle indicates the load type being transported. The height of a rectangle corresponds to the number of bots involved in the poly-robot. Without reconfiguration, the optimal number of bots is $N_W = 6$. With reconfiguration, the optimal number of bots is $N_R = 4$.

Configuration	Load type			
	$k = 1$ (S)	$k = 2$ (M)	$k = 3$ (L)	$k = 4$ (XL)
1-bot	1	0	0	0
2-bot	1	1	0	0
3-bot	1	1	1	0
4-bot	1	1	1	1

Table 1: Capacities for an example with 4 types of loads and configurations



(a) Without reconfiguration

(b) With reconfiguration

Figure 3: Gantt chart for an example with four types of loads

We briefly comment the solutions presented in Figure 6. When reconfiguration is not allowed, the 4-bot is not able to reconfigure, so it carries a single XL load, then a single L load, a single M load and a single S load. It follows that we need two additional 1-bots to carry the remaining S loads. When reconfiguration is allowed, a 4-bot carries one load XL in period 1. Then it reconfigures in period 2 into a 3-bot plus a 1-bot. In period 3, the 3-bot reconfigures itself into a 2-bot plus a 1-bot. Finally, in period 4, the 2-bot reconfigures into two 1-bots.

Discussion of assumptions. The MecaBotiX company envisions a practical industrial application where various load types correspond to materials stored in distinct zones of a large warehouse. A typical transportation task for a load of type k involves collecting a number of items n ($n \leq c_{pk}$) from a pick-up and delivery platform A_0, loading them onto a p-bot, delivering them to the designated storage area A_k, and then returning to A_0. Alternatively, the task might require the p-bot to move to A_k, pick up loads of type k, and transport them back to A_0.

Each transportation task typically lasts between 10 and 15 minutes, part of which is allocated for handling the load within the target area A_k. However, the exact duration depends on factors like warehouse size, which can range from a few thousand to over a million square feet.

The process of reconfiguring a poly-bot, performed by warehouse staff, is estimated to take less than two minutes and usually occurs between consecutive time periods. In practice, reconfiguration is expected to be rare, as a p-bot assigned to transport type k loads during period t will likely continue the same task in the subsequent period $t+1$. With advancements in poly-robot industrialization, reconfiguration times are anticipated to become negligible.

Managing uncertainties in trip durations, reconfiguration, and load handling remains challenging. Dividing time into structured periods helps synchronize transportation tasks with worker schedules and external operations (e.g., production or logistics). Period lengths are expected to range from 10 to 20 minutes, depending on warehouse context and size. Over a typical workday, approximately 40 such periods can be accommodated. Transportation tasks should be completed within a single period, ensuring synchronization with other processes.

4. Mathematical formulations

In this section, we present Integer Linear Program (ILP) formulations for the two variants.

4.1. Without reconfiguration

We first assume that reconfiguration is prohibited. We denote by x_p the decision variable representing the number of p-bots chosen for the entire horizon. The total number of bots is then $\sum_{p=1}^{P} p \cdot x_p$. We also denote by x_{pk}^t the decision variable representing the number of poly-robots in configuration p transporting loads of type k in period t. For the solution illustrated in Figure 3a, the decision variables that are not equal to zero are: $x_1 = 2, x_4 = 1, x_{11}^1 = 2, x_{44}^1 = 1, x_{11}^2 = 2, x_{43}^2 = 1, x_{11}^3 = 1, x_{42}^3 = 1, x_{11}^4 = 1, x_{41}^4 = 1$. The number of required bots is then $x_1 + 4x_4 = 6$.

The problem without reconfiguration can be formulated by the following ILP.

$$N_W = \min \quad \sum_{p=1}^{P} p \cdot x_p$$

$$\text{subject to :}$$

$$\sum_{t=1}^{T} \sum_{p=1}^{P} c_{pk} \cdot x_{pk}^t \geq d_k \qquad \forall k \tag{1}$$

$$x_p \geq \sum_{k=1}^{K} x_{pk}^t \qquad \forall t, \forall p \tag{2}$$

$$x_p \in \mathbb{N},\, x_{pk}^t \in \mathbb{N} \qquad \forall k, \forall p, \forall t$$

Constraint (1) means that the total capacity of the fleet along the time horizon must be able to transport all loads of each type. Constraint (2) means that the number x_p of p-bots must be greater than or equal to the number of p-bots used over each period.

4.2. With reconfiguration

When reconfiguration is allowed, the problem can be formulated by the following ILP. We remind that x^t_{pk} is the decision variable representing the number of poly-robots in configuration p transporting loads of type k in period t.

$$
\begin{aligned}
N_R = \min\ & N \\
& \text{subject to :} \\
& \sum_{t=1}^{T} \sum_{p=1}^{P} c_{pk} \cdot x^t_{pk} \geq d_k && \forall k && (3) \\
& N \geq \sum_{k=1}^{K} \sum_{p=1}^{P} p \cdot x^t_{pk} && \forall t && (4) \\
& N \in \mathbb{N},\ x^t_{pk} \in \mathbb{N} && \forall k,\ \forall p,\ \forall t
\end{aligned}
$$

Constraint (3) is the same as in the problem without reconfiguration. Constraint (4) means that the number of bots used, N, must be greater than or equal to the number of bots used over each period.

Note that the problem with reconfiguration is strongly NP-hard, even when there is a single configuration $p(k)$ for each load type k. This special case is precisely an IMS (Identical Machine Scheduling) problem, denoted as $Pm||C_{max}$ in the scheduling literature (Pinedo, 2012). The correspondence between our problem (with a single configuration per type of load) and the IMS problem is as follows. The T periods correspond to T parallel machines. The K load types correspond to K different types of jobs. The demand d_k corresponds to d_k jobs of type k with processing time $p(k)$. Since the IMS problem is strongly NP-hard (Graham, 1969), our problem with reconfiguration is also strongly NP-hard.

4.3. Additional operational constraints

We can add various operational constraints to the basic ILP formulations. For instance, we can impose a maintenance of the fleet in period t by setting $x^t_{pk} = 0$ for all p and k. We can also impose that demands of type k must be satisfied in a time window $\mathcal{T}_k \subset \{1, \cdots, T\}$, by letting $x^t_{pk} = 0$ for all $t \notin \mathcal{T}_k$ and all p. To take into account failures, we can introduce an efficiency $e \in]0, 1]$ of vehicles after accounting for breakdowns (Egbelu, 1987), where $e = 1$

means that there is no failure. We should then set the number of bots to $\lceil N_R/e \rceil$ and $\lceil N_W/e \rceil$ for the problems with or without reconfiguration.

5. Minimum number of bots for the special case of unit capacities

We now consider K types of loads and $P = K$ configurations with 0-1 capacities. More precisely, the capacity matrix is lower triangular and is such that c_{kp} is equal to 1 if $p \geq k$ and 0 otherwise. We also assume that there is at least one load of type K ($d_K \geq 1$), otherwise the problem reduces to a problem with $(K-1)$ types of loads.

Table 2 illustrates the capacity matrix with $K = 3$ types of loads and $P = 3$ configurations. We can imagine that loads of type 1, 2, 3 correspond to respectively Small (S), Medium (M) and Large (L) loads. With this terminology, small loads can be transported by all configurations, medium loads by 2-bots or 3-bots and large loads only by 3-bots.

Configuration	Load type		
	$k=1$ (S)	$k=2$ (M)	$k=3$ (L)
1-bot	1	0	0
2-bot	1	1	0
3-bot	1	1	1

Table 2: Capacities for 3 types of loads

When considering such capacity structure, we are able to derive simple formulas for the minimum number of bots with or without reconfigurations. In the case without reconfiguration, we obtain a result that holds for an arbitrary number of types of loads K. In the case with reconfiguration, we obtain a result up to 3 types of loads. With 4 types of loads or more, the problem becomes much more complex as the reconfiguration decision is no more trivial. For instance, we can reconfigure a 4-bot into two 2-bots or into a 3-bot plus a 1-bot.

Theorem 1 (Unit capacities) *Let $d'_K = 0$ and for $k = K, \cdots, 2$*

$$x_k = \left\lceil \frac{(d_k - d'_k)^+}{T} \right\rceil$$

$$d'_{k-1} = d'_k - d_k + x_k \cdot T$$

Then

$$N_W = \sum_{k=1}^{K} k \cdot x_k \tag{5}$$

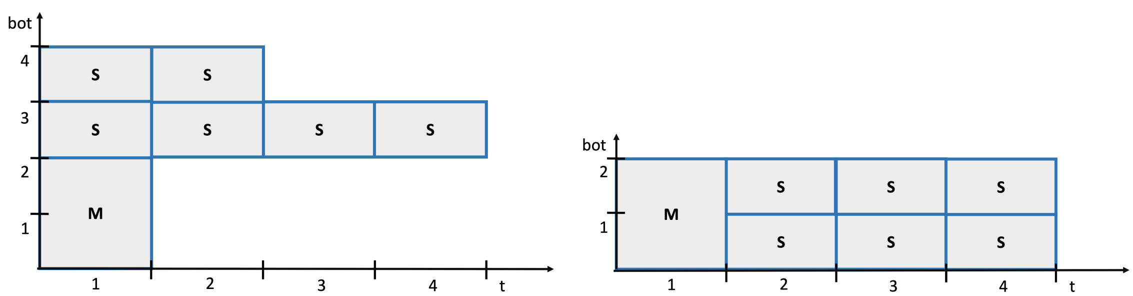
and, for $K \leq 3$,

$$N_R = \sum_{k=2}^{K} k \cdot x_k + \left\lceil \frac{(d_1 - d'_1 - \sum_{k=1}^{K-1} d'_k)^+}{T} \right\rceil \tag{6}$$

The proof of this theorem is detailed in Appendix A. Note that (6) holds for $K > 3$ if we further assume that there is a sufficient number of type 1 loads to exploit the partially used slots (e.g. $d_1 \geq K(T-1)$). In this theorem, x_k represents the number of required k-bots and d'_k the number of free periods in the last used configuration after the assignment of loads of type $k+1, \cdots, K$.

6. Reconfigurable versus non-reconfigurable fleet

We now examine how many bots can be saved thanks to reconfigurability. To begin with, we present a simple example with two types of loads where the fleet size can be divided by two. In this example, loads of type 1 and 2 correspond to respectively small (S) and medium (M) loads. Let $T = 4$, $P = K = 2$, $d_1 = 6$, $d_2 = 1$, $c_{11} = c_{22} = 1$ and $c_{12} = c_{21} = 0$. Then $N_R = 2$, $N_W = 4$ and $\frac{N_W}{N_R} = 2$ (see Figure 4 for the Gantt chart).



(a) Without reconfiguration (b) With reconfiguration

Figure 4: Gantt chart of an example where the limit is reached with two types of loads

We prove in Appendix B that we can remove at most PK bots thanks to reconfigurability.

Theorem 2 (General case) *For the problem described in Section 3, we have*

$$0 \leq N_W - N_R \leq PK \tag{7}$$

Equation (7) can be rewritten as

$$\frac{N_W}{N_R} \leq 1 + \frac{PK}{N_R}.$$

It implies that the potential gain is diminishing with the fleet size. For instance, let $P = K = 4$. If $N_R = 10$, then $\dfrac{N_W}{N_R} \leq 2.6$. If $N_R = 100$, then $\dfrac{N_W}{N_R} \leq 1.16$. If $N_R = 1000$, then $\dfrac{N_W}{N_R} \leq 1.016$.

6.1. Unit capacities

For the setting with unit capacities (described in Section 5), we now compare the optimal numbers of bots with or without reconfiguration. As formula (6) in Theorem 1 holds only for $K \leq 3$, we assume in that there are at most 3 load types.

Theorem 3 (Unit capacities) *When $K \leq 3$, we have*

$$0 \leq N_W - N_R \leq K - 1. \tag{8}$$

The proof of this theorem can be found in Appendix C. This theorem shows that the gain is not substantial in absolute value. When $K = 1$, i.e. when there is a single load type, the gain is null as expected. When $K = 2$, we can gain at most one bot by using reconfigurability. When $K = 3$, the gain is at most of 2 bots. Note that Theorem 3 holds for $K > 3$ if we further assume that there is sufficient number of loads of type 1 to exploit the partially used slots.

Theorem 3 implies that $\frac{N_W}{N_R} \leq 1 + \frac{K-1}{N_R}$. As $d_K \geq 1$, we have $N_R \geq K$ and we get that

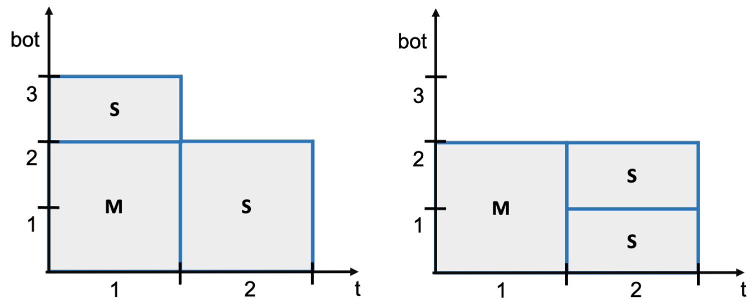
$$1 \leq \frac{N_W}{N_R} \leq 2 - \frac{1}{K}. \tag{9}$$

In what follows, we provide examples where the upper bound in (8) and (9) are reached for two or three types of loads.

Example with two types of loads

In the following example, loads of type 1 and 2 correspond to respectively small (S) and medium (M) loads. Let's take $T = 2$, $d_1 = 2$ and $d_2 = 1$. Then $N_R = 2$ and $N_W = 3$ and $\frac{N_W}{N_R} = \frac{3}{2}$. Figure 5 represents the Gantt chart for this example. On the ordinate axis is the reference number of the bots and on the abscissa axis is the reference number of the period. In each rectangle is indicated the carried load type.

In case where reconfiguration is permitted, the 2-bot can transport one medium load before transforming into two 1-bots, which are then capable of carrying two small loads. On the other hand, if reconfiguration is not allowed, the 2-bot cannot divide into two independent robots, and can only transport a single small load.

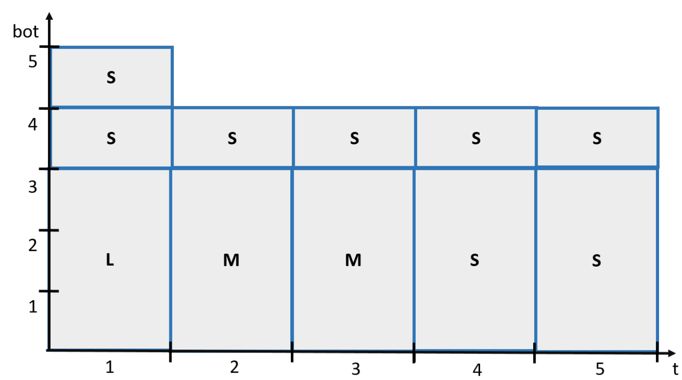


(a) Without reconfiguration (b) With reconfiguration

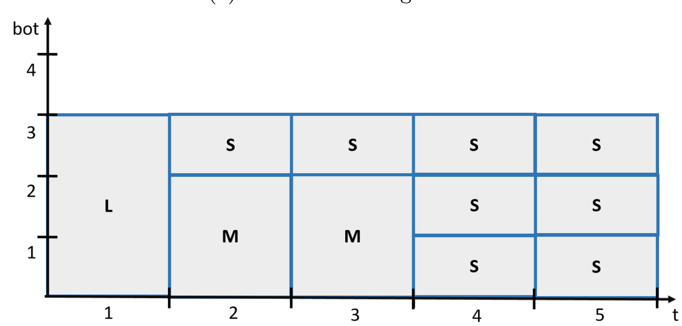
Figure 5: Gantt chart for an example with two types of loads

Example with three load types

In the following example, loads of type 1, 2 and 3 correspond to respectively small (S), medium (M) and large (L) loads. Let's take $T = 5$, $d_1 = 8$, $d_2 = 2$ and $d_3 = 1$. Then $N_R = 3$, $N_W = 5$ and $\frac{N_W}{N_R} = \frac{5}{3}$. Figure 6 represents the Gantt chart of an optimal solution.



(a) Without reconfiguration



(b) With reconfiguration

Figure 6: Gantt chart for an example with three load types

6.2. Demand per period

So far, we have shown that the gain in relative value, N_W/N_R, can be significant while the gain in absolute value, $N_W - N_R$, remains limited. We will now consider a variation of the problem where the gain in absolute value can be significant.

We now assume that the demand is per period and not over the whole horizon. More precisely, there are d_{kt} loads of type k to be transported in period t (instead of d_k loads of type k to be transported over the whole horizon in the previous sections). This may correspond, for example, to the activity of a warehouse that delivers one type of load on Monday and another type of load on Tuesday. With this new assumption, the ILPs remain unchanged except that constraints (1) and (3) have to be replaced by the following one:

$$\sum_{p=1}^{P} c_{pk} \cdot x_{pk}^{t} \geq d_{kt} \qquad \forall k, \forall t \tag{10}$$

We present below a simple example where the gain in absolute value can be significant. Consider two types of loads and two periods where d_1 loads of type 1 have to be transported in period 1 and d_2 loads of type 2 have to be transported in period 2 (see Table 3a for a summary of demands). Type 1 (S) loads can only be transported by 1-bots while type 2 (M) loads can only be transported by p-bots with $p \geq 2$ (see Table 3b for a summary of capacities).

Load type	Period	
	$t = 1$	$t = 2$
$k = 1$ (S)	d_1	0
$k = 2$ (M)	0	d_2

(a) Demands

Configuration	Load type	
	$k = 1$ (S)	$k = 2$ (M)
1-bot	1	0
p-bot	0	1

(b) Capacities

Table 3: Instance where the gain in absolute value can be significant

For this example, the minimum numbers of bots is trivial and can be expressed as:

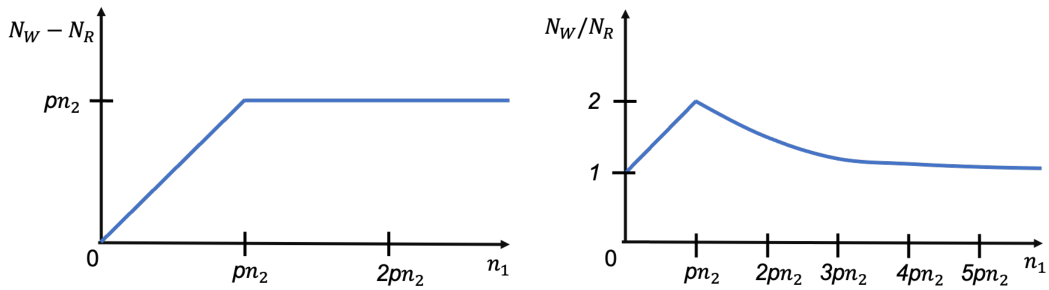
$$\begin{aligned} N_W &= d_1 + pd_2, \\ N_R &= (d_1 - pd_2)^+ + pd_2 \\ &= \max(d_1, pd_2). \end{aligned}$$

It follows that

$$N_W - N_R = \min(d_1, pd_2),$$

$$\frac{N_W}{N_R} = \begin{cases} 1 + \dfrac{d_1}{pd_2} & \text{if } d_1 \leq pd_2 \\ 1 + \dfrac{pd_2}{d_1} & \text{if } d_1 > pd_2. \end{cases}$$

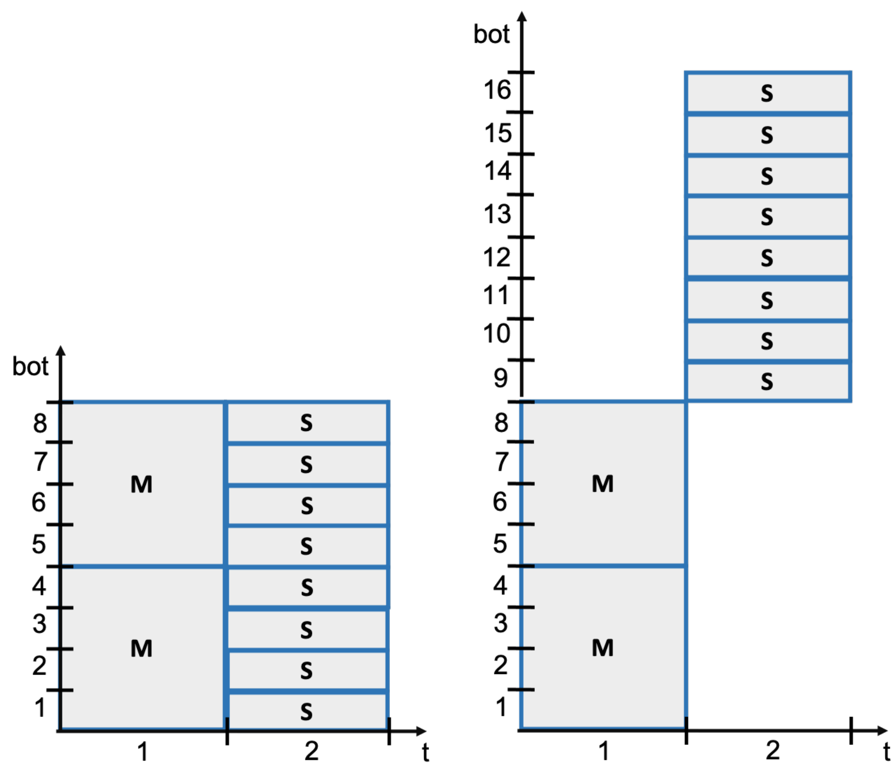
When $d_1 = pd_2$, the ratio N_W/N_R is maximum and equal to 2 while the difference is equal to d_1 and is thus unbounded. Figure 7 plots the effect of d_1 on the difference $N_W - N_R$ and the ratio $\frac{N_W}{N_R}$.



(a) Difference (b) Ratio

Figure 7: Effect of the number of loads of type 1 on the gains in absolute and relative value

Figure 8 represents the Gantt chart for the above example with $p = 4$, $d_1 = 8$ and $d_2 = 2$. We have then $N_R = 8$ and $N_W = 16$, $N_W - N_R = 8$ and $N_W/N_R = 2$.



(a) With reconfiguration (b) Without reconfiguration

Figure 8: Gantt chart for the instance described in Table 3 with $p = 4$, $d_1 = 8$ and $d_2 = 2$

7. Numerical experiments

In this section, we solve the two ILPs introduced in Section 4 with the PuLP library, using CBC (Coin Branch and Cut) solver. The programs are implemented in Python on a PC AMD Opteron 2.1GHz.

Instances. We build an instance by first choosing T, P, K and next generating the capacities and the demands as follows. For each load type k, we randomly select the capacity c_{1k} in $\{1, \cdots, K\}$. Then we set, $c_{pk} = p \cdot (c_{1k} + \epsilon) + \tau$ with ϵ randomly selected in $\{1, 2\}$ and τ randomly selected in $\{1, 3\}$ for each configuration $p > 1$. We denote by $c_k^{mean} = \frac{1}{P} \sum_p c_{pk}$ the mean capacity. The demand for load type k is then set as $d_k = \lfloor \gamma \cdot J \cdot T \cdot P \cdot c_k^{mean} \rfloor$ where J is randomly selected in $\{1, \cdots, 10\}$. Note that demand d_k increases with the demand factor γ.

We call nominal instance the instance with parameters ($T = 10$, $P = 5$, $K = 5$, $\gamma = 0.1$). We first apply the ILP programs to this nominal instance, then change one parameter at a time among T, P, K, γ, while keeping the others unchanged, as follows:

- $T = 10, 20, 30, 40, 80, 120$
- $K = 2, 3, 5, 6, 9, 12$
- $P = 2, 3, 5, 6, 9, 12$
- $\gamma = 0.1, 0.5, 1, 5, 10$

Results. The solver efficiently computes optimal or near-optimal solutions in less than one second for the scenarios involving $K \leq 4$ types of loads, $P \leq 4$ configurations and $T \leq 40$ periods. However, as the parameters K, P and T grow, the computation time increases significantly. Table 4 provides the CPU times to obtain the optimal solution for different values of T. For each value of T, we generate 10 different instances and compute the average CPU time for the instances solved to optimality. Note that the difficult instances are not the same for the problem with reconfiguration or without.

T	Average CPU time without reconfiguration (*)	Number of instances solved to optimality without reconfiguration	Average CPU time with reconfiguration (*)	Number of instances solved to optimality with reconfiguration
10	2.31 s.	9	0.24 s.	9
20	1.21 s.	10	2.97 s.	10
40	0.71 s.	9	6.26 s.	8
80	1.08 s.	9	2.68 s.	5
160	3.84 s.	9	7.84 s.	2

Table 4: Effect of the number of periods T on the computation time (with $P = 5$, $K = 5$, $\gamma = 0.1$)
(*) for instances solved to optimality in less than 10 minutes

For large instances, solving the ILP becomes impractical within a reasonable timeframe, necessitating the development of heuristic approaches. Fortunately, in real-world applications, most warehouses typically handle only a limited number of standardized load types, such as boxes and pallets. To handle larger instances, it would be interesting to design a simple and efficient heuristic algorithm. To design such algorithm, we could take advantage of the fact that our problem shares characteristics with the IMS problem (see Section 4.2). When there is a single configuration available for each type of load, our problem is precisely an IMS problem that could be solved by well known heuristics such as LPT (Longest Processing Time First). One possible heuristic scheme would the following. First choose, for each type of load, the most efficient configuration, i.e. the configuration that has a maximum capacity per bot. Second, solve the IMS problem. Third, try to improve the solution by using other configurations.

8. Conclusion

Just like human beings for whom it is natural to carry small loads alone and to collaborate with others to carry heavy or bulky loads, we consider a fleet of reconfigurable robots that adapt themselves according to the load type to be carried. We present the problem of sizing such a fleet of reconfigurable robots and propose a mathematical formulation. We show that the resulting ILP models can be solved in a very short computation time with a limited number of load types and configurations, a situation that corresponds to many warehouse settings. For the special case where loads are carried one by one, we derive closed-form expressions for the minimum number of robots up to three load types with reconfiguration, and for any number of load types without reconfiguration. We also investigate how reconfigurability can reduce the number of bots needed. We show that the value of reconfigurability can be very high but diminishes with the fleet size. We also observe that reconfigurability can be particularly useful when the demand for small loads has to be met at a different time interval from the demand for large loads.

A number of simplifying assumptions have been made, which could be relaxed in future work. One potential direction would be to explore more complex warehouse topologies or transportation times that depend on the type of load and configuration. Another interesting avenue would involve integrating reconfiguration time and costs in the model, as well as accounting for stochastic demands and transportation times. Additionally, examining polyrobots whose capacities are influenced by the geometry of their configuration, rather than solely by their number, could provide valuable insights. In practice, the transportation process in an industrial warehouse is dynamic, and we have to deal with new demands and random events that force decision makers to frequently update the planning of transportation tasks. Hence, we need to be able to generate good feasible solutions in a very short time. Designing a fast and efficient heuristic algorithm that would be able to tackle large instances is therefore an area for research.

Appendix A. Proof of Theorem 1

Assume in all this proof that $K = P$ and that c_{kp} is equal to 1 if $p \geq k$ and 0 otherwise.

Appendix A.1. Without reconfiguration

We begin by exploring the case without reconfiguration. Note that, if you have a p-bot, it is optimal to use it in priority to transport the biggest loads (with the highest index). Remind that we denote by x_k the minimum number of required k-bots. In what follows, we determine $x_k, x_{k-1}, \cdots, x_1$ in this order.

We also denote by d'_k the number of free periods in the last used configuration after the assignment of loads of type $k+1, \cdots, K$. It is optimal to use these free periods in priority for loads of type k, then of type $k-1$ and so on. Figure A.9 illustrates notation d'_k. In this figure, $d'_3 = 4$ means that we have 4 free periods for the transport of the loads of type 3 (L). $d'_2 = 2$ means that we have 2 free periods for the transport of loads of type 2 (M). $d'_1 = 3$ means that we have 3 free periods for the transport the loads of type 1 (S).

The number of K-bots necessary to transport loads of type K is

$$x_K = \left\lceil \frac{d_K}{T} \right\rceil .$$

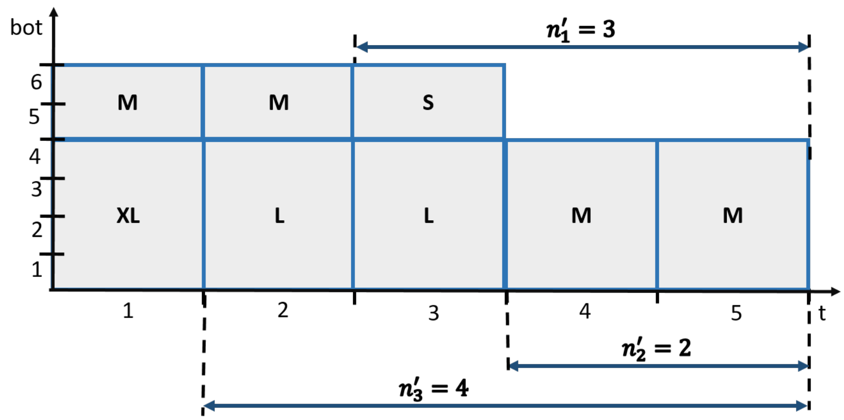


Figure A.9: Illustration of notation d'_k with four types of loads ($T = 5, d_1 = 1, d_2 = 4, d_3 = 2, d_4 = 1$)

It then may remain available capacity for the last K-bot. More precisely, the number of free periods for the last K-bot is

$$d'_{K-1} = x_K \cdot T - d_K = \left\lceil \frac{d_K}{T} \right\rceil T - d_K.$$

These d'_{K-1} free periods are used to transport in priority loads of type $(K-1)$ and it remains to transport $(d_{K-1} - d'_{K-1})^+$ loads of type $K-1$. The number of $(K-1)$-bots necessary to transport these remaining loads of type $(K-1)$ is then

$$x_{K-1} = \left\lceil \frac{(d_{K-1} - d'_{K-1})^+}{T} \right\rceil.$$

Assume now that the number of p-bots, x_p, has been determined for $p = k+1, \cdots, K$ and that there remains d'_k free periods on these configurations. These d'_k free periods are assigned in priority to loads of type k and then $(d_k - d'_k)^+$ loads of type k remain to be carried. The number of additional k-bots required is

$$x_k = \left\lceil \frac{(d_k - d'_k)^+}{T} \right\rceil.$$

The number of free periods for the last k-bot is $x_k \cdot T - (d_k - d'_k)^+$. There remains also $(d'_k - d_k)^+$ free periods after the transportation of loads of type j (for $j > k$). It follows that

$$\begin{aligned} d'_{k-1} &= x_k \cdot T - (d_k - d'_k)^+ + (d'_k - d_k)^+ \\ &= x_k \cdot T + d'_k - d_k \end{aligned}$$

In the end, we have

$$N_W = \sum_{k=1}^{K} k \cdot x_k$$

with

$$\begin{aligned} x_k &= \left\lceil \frac{(d_k - d'_k)^+}{T} \right\rceil \\ d'_K &= 0 \\ d'_{k-1} &= d'_k - d_k + x_k \cdot T \qquad \text{(for } k = K, \cdots, 2) \end{aligned}$$

Appendix A.2. With reconfiguration

<u>$K = 1$:</u> With a single load type, we have immediately $N_R = \lceil \frac{d_1}{T} \rceil$ and it is easy to check that (6) gives the same result.

<u>$K = 2$:</u> When there are two types of loads, we need x_2 2-bots, as in the case without reconfiguration. If there remains free periods on the last 2-bot, it is optimal to reconfigure into two 1-bots. On the d'_1 free periods of the last 2-bot, we can transport up to $2d'_1$ loads of type 1. It remains $(d_1 - 2d'_1)^+$ loads of type 2 that require $\left\lceil \frac{(d_1-2d'_1)^+}{T} \right\rceil$ additional 1-bots. In the end, the optimal number of bots is

$$N_R = 2 \cdot x_2 + \left\lceil \frac{(d_1 - 2d'_1)^+}{T} \right\rceil$$

and we have shown that (6) holds.

<u>$K = 3$:</u> Let's detail now the case with 3 types of loads. We need x_3 3-bots to transport loads of type 3, as in the case without reconfiguration. If there remains free periods on the last 3-bot, it is optimal to reconfigure into a 2-bot plus a 1-bot. On the d'_2 free periods of the last 3-bot, we can then transport up to d'_2 loads of type 2 and d'_2 loads of type 1. It remains $(d_2 - d'_2)^+$ loads of type 2 that require $x_2 = \left\lceil \frac{(d_2-d'_2)^+}{T} \right\rceil$ additional 2-bots, as in the case without reconfiguration. On the last 2-bots, there are d'_1 free periods which can be used to transport up to $2d'_1$ loads of type 1. It remains $(d_1 - d'_2 - 2d'_1)$ loads of type 1 that require $\left\lceil \frac{(d_1-d'_2-2d'_1)^+}{T} \right\rceil$ additional 1-bots. If ever $d_2 = 0$, then the 3-bot reconfigures into three 1-bots and $d'_2 = d'_1$. In the end, we have

$$N_R = 3 \cdot x_3 + 2 \cdot x_2 + \left\lceil \frac{(d_1 - d'_2 - 2d'_1)^+}{T} \right\rceil$$

and we have shown that (6) holds.

Appendix B. Proof of Theorem 2

Let $p_0(k) = \arg\max_p \frac{c_{pk}}{p}$ the most profitable configuration. Let also $x_k = \left\lfloor \frac{d_k}{Tc_{p_0(k),k}} \right\rfloor$ and $x'_k = \left\lceil \frac{d_k}{Tc_{p_0(k),k}} \right\rceil$. On one hand, we have $N_R \geq \sum_{k=1}^{K} x_k p_0(k)$. On the other hand, we have $N_W \leq \sum_{k=1}^{K} x'_k p_0(k)$. It follows that $N_W - N_R \leq \sum_{k=1}^{K} p_0(k)(x'_k - x_k)$. Since $p_0(k) \leq P$ and $x'_k - x_k \leq 1$, it follows that $N_W - N_R \leq PK$.

Appendix C. Proof of Theorem 3

From (5) and (6), we obtain that

$$N_W - N_R = \tag{C.1}$$

$$= \left\lceil \frac{(d_1 - d'_1)^+}{T} \right\rceil - \left\lceil \left(\frac{d_1 - d'_1}{T} - \frac{1}{T} \cdot \sum_{k=1}^{K-1} d'_k \right)^+ \right\rceil . \tag{C.2}$$

Note that the number of free periods d'_k can not exceed $T-1$ as there is at least one period used to transport some load. Thus $d'_k \leq T-1$ for $k = 1, \cdots, K-1$. It follows that

$$N_W - N_R \leq \left\lceil \frac{(d_1 - d'_1)^+}{T} \right\rceil - \left\lceil \left(\frac{d_1 - d'_1}{T} - (K-1) \right)^+ \right\rceil . \tag{C.3}$$

To conclude we need the following property. Let b a non-negative integer and x a real number. Then $\lceil x^+ \rceil - \lceil (x-b)^+ \rceil \leq b$. The proof of this property is straightforward.

$$\lceil x^+ \rceil - \lceil (x-b)^+ \rceil = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x \leq b \\ b & \text{if } x > b \end{cases}$$
$$\leq b$$

Using this property and (C.3) gives $N_W - N_R \leq K - 1$.

References

M. Amjath, L. Kerbache, A. Elomri, and J. M. Smith. Fleet sizing of heterogeneous fleet of trucks in a material handling system using anylogic simulation modelling. *Proceedings of the 5th European International Conference on Industrial Engineering and Operations Management*, 2022.

I. Aziez, J.-F. Côté, and L. C. Coelho. *Fleet Sizing of Healthcare Automated Guided Vehicles*. Bureau de Montreal, Université de Montreal, 2021.

A. Baykasoğlu, K. Subulan, A. S. Taşan, and N. Dudaklı. A review of fleet planning problems in single and multimodal transportation systems. *Transportmetrica A: Transport Science*, 15(2):631–697, 2019.

H. Bojinov, A. Casal, and T. Hogg. Emergent structures in modular self-reconfigurable robots. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, volume 2, pages 1734–1741, 2000.

D. Brandt, D. J. Christensen, and H. H. Lund. Atron robots: versatility from self-reconfigurable modules. In *2007 International Conference on Mechatronics and Automation*, pages 26–32. IEEE, 2007.

A. Castano, R. Chokkalingam, and P. Will. Autonomous and self-sufficient conro modules for reconfigurable robots. In *Distributed Autonomous Robotic Systems 4*, pages 155–164. Springer, 2000.

Z. E. Chebab. *Conception et commande collaborative de manipulateurs mobiles modulaires (C3M3)*. PhD thesis, Université Clermont Auvergne, 2018.

M.-S. Cheon, K. C. Furman, and D. Shaffer, Timothy. A modeling framework for railcar fleet sizing in the chemical industry. *Industrial & engineering chemistry research*, 51(29): 9825–9834, 2012.

F. F. Choobineh, A. Asef-Vaziri, and X. Huang. Fleet sizing of automated guided vehicles: a linear programming approach based on closed queuing networks. *International Journal of Production Research*, 50(12):3222–3235, 2012.

F. Cui, Z. Jiang, X. Zhou, J. Zheng, and N. Geng. A configuration optimization approach for reconfigurable manufacturing system based on column-generation combined with graph neural network. *International Journal of Production Research*, pages 1–22, 2024.

P. J. Egbelu. The use of non-simulation approaches in estimating vehicle requirements in an automated guided based transport system. *Material flow*, 4(1-2):17–32, 1987.

J. L. Everett, A. C. Hax, V. A. Lewinson, and D. Nudds. Optimization of a fleet of large tankers and bulkers: A linear programming approach. *Marine Technology and SNAME News*, 9(04):430–438, 1972.

G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, 294(2):405–426, 2021.

J. Gödeke and P. Detzner. A simulative approach to amr fleet sizing in decentralized multi-robot task allocation. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2023.

R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2):416–429, 1969.

P. Guo, Z. Chen, Y. Yang, and R. Miao. A multistage simulation-optimization-integrated methodology framework for user-oriented electric vehicle carsharing reallocation under dynamic price subsidy. *Energy*, 290:130207, 2024a.

P. Guo, Y. Yang, Y. Su, Z. Chen, and R. Miao. Incentive-based customer-oriented rebalancing strategy for one-way shared electric vehicles in sustainable urban governance. *Journal of Cleaner Production*, 469:143192, 2024b.

A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010.

J. Irawan, X. De Kestelier, N. Argyros, B. Lewis, and S. Gregson. A reconfigurable modular swarm robotic system for isru (in-situ resource utilisation) autonomous 3D printing in extreme environments. In *Design Modelling Symposium Berlin*, pages 685–698. Springer, 2019.

M. R. Jahanshahi, W.-M. Shen, T. G. Mondal, M. Abdelbarr, S. F. Masri, and U. A. Qidwai. Reconfigurable swarm robots for structural health monitoring: a brief review. *International Journal of Intelligent Robotics and Applications*, 1:287–305, 2017.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

P.-H. Koo, J. Jang, and J. Suh. Estimation of part waiting time and fleet sizing in agv systems. *International journal of flexible Manufacturing Systems*, 16(3):211–228, 2004.

A. Kumar, D. Roy, V. Verter, and D. Sharma. Integrated fleet mix and routing decision for hazmat transportation: A developing country perspective. *European Journal of Operational Research*, 264(1):225–238, 2018.

M. Lai and H. K. Lo. Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A: Policy and Practice*, 38(4):305–328, 2004.

H.-Y. Lee and C. C. Murray. Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research*, 57(18):5821–5841, 2019.

P. Y. Leong and N. S. Ahmad. Exploring autonomous load-carrying mobile robots in indoor settings: A comprehensive review. *IEEE Access*, 2024.

X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo. Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems. *IEEE Access*, 7:74909–74924, 2019.

J. A. Manzolli, J. P. Trovão, and C. H. Antunes. A review of electric bus vehicles research topics–methods and trends. *Renewable and Sustainable Energy Reviews*, 159:112211, 2022.

MecaBotiX. `https://www.mecabotix.com/`, 2024.

A. Mehrez, M. S. Hung, and B. H. Ahn. An industrial ocean-cargo shipping problem. *Decision Sciences*, 26(3):395–423, 1995.

Q. Meng and T. Wang. A scenario-based dynamic programming model for multi-period liner ship fleet planning. *Transportation Research Part E: Logistics and Transportation Review*, 47(4):401–413, 2011.

R. Miao, Q. Li, W. Huang, P. Guo, L. Mi, Z. Zhang, J. Zhang, and Z. Jiang. Profit optimization for mileage-based pricing of electric vehicle lease. *IEEE Transactions on Engineering Management*, 69(4):951–962, 2020.

M. Milenković and N. Bojović. A fuzzy random model for rail freight car fleet sizing problem. *Transportation Research Part C: Emerging Technologies*, 33:107–133, 2013.

M. Mohtasham, H. Mirzaei-Nasirabad, H. Askari-Nasab, and B. Alizadeh. Truck fleet size selection in open-pit mines based on the match factor using a minlp model. *Mining Technology*, 130(3):159–175, 2021.

E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund. Design of the atron lattice-based self-reconfigurable robot. *Autonomous Robots*, 21:165–183, 2006.

G. Pantuso, K. Fagerholt, and L. M. Hvattum. A survey on maritime fleet size and mix problems. *European Journal of Operational Research*, 235(2):341–349, 2014.

M. L. Pinedo. *Scheduling*, volume 29. Springer, 2012.

M. A. Post, X.-T. Yan, and P. Letier. Modularity for the future in space robotics: A review. *Acta Astronautica*, 189:530–547, 2021.

A. Rahimi-Vahed, T. G. Crainic, M. Gendreau, and W. Rei. Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm. *Computers & Operations Research*, 53:9–23, 2015.

N. Schwartz. Discrete programs for moving known cargos from origins to destinations on time at minimum bargeline fleet cost. *Transportation Science*, 2(2):134–145, 1968.

J. Seo, J. Paik, and M. Yim. Modular reconfigurable robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:63–88, 2019.

M. Sha and R. Srinivasan. Fleet sizing in chemical supply chains using agent-based simulation. *Computers & Chemical Engineering*, 84:180–198, 2016.

W.-m. Shen, M. Krivokon, H. C. H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh. Multimode locomotion via superbot reconfigurable robots. *Auton. Robots*, 20:165–177, 03 2006. doi: 10.1007/s10514-006-6475-7.

Z. Soufi, S. Mestiri, P. David, Z. Yahouni, and J. Fottner. A material handling system modeling framework: a data-driven approach for the generation of discrete-event simulation models. *Flexible Services and Manufacturing Journal*, pages 1–30, 2024.

K. Stoy, D. Brandt, D. J. Christensen, and D. Brandt. Self-reconfigurable robots: an introduction. 2010.

O. Tellez, S. Vercraene, F. Lehuédé, O. Péton, and T. Monteiro. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 91:99–123, 2018.

I. F. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3):677–709, 2006.

A. Y. S. Wan, A. V. Le, C. G. Moo, V. Sivanantham, and M. R. Elara. Enabling framework for constant complexity model in autonomous inter-reconfigurable robots. *IEEE Transactions on Automation Science and Engineering*, 2024.

X. Xinlian, W. Tengfei, and C. Daisong. A dynamic model and algorithm for fleet planning. *Maritime Policy & Management*, 27(1):53–63, 2000.

M. Yim, D. G. Duff, and K. D. Roufas. Polybot: a modular reconfigurable robot. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, volume 1, pages 514–520, 2000.