"Where do I stack it?" A review of the heuristics for the block relocation problem in open yards

Lagorio, A. *, Pinto, R.*

** Department of Management, Information and Production Engineering, University of Bergamo, Viale Marconi 5, 24044 – Dalmine – Italy (chiara.cimini@unibg.it, alexandra.lagorio@unibg.it, roberto.pinto@unibg.it)*

Abstract: The management of inventory movements in open yard storage entails many aspects of both the type of unit loads handled (usually containers, swap bodies or large and heavy pallet) and the sequence of arrival and retrieval. In this highly dynamic scenario, the correct location of incoming load units in the yard is essential to ensure the effective use of resources (such as cranes and forklifts) and the timely shipment of the orders to the customers. In this respect, one of the main goals in managing open yard storage is to minimise handling operations. In particular, it is necessary to limit reshuffle operations, i.e. load units movement operations inside the yard to allow access to other units. In literature, a solution to this problem is represented by the research on the Block Relocation Problem (BRP), mainly developed in the seaport context. Both analytic and heuristics approaches have been formulated with different characteristics, levels of difficulty and effectiveness. Motivated by research activity in a company, in this paper, we develop a review of the main existing heuristics for solving BRPs, considering the importance they can play in yards management to optimise handling operations, reducing the number of movements.

Keywords: Block Relocation Problem, storage yard management, heuristics.

1. Introduction

Open yard storage (i.e., the storage of containers, swap bodies or pallets of large products in an open area) usually requires dedicated handling equipment (e.g., gantries, cranes, and special forklifts capable of handling many tonnes) as well as suitable management policies. Such policies, in particular, need to reflect the specific characteristics of the storage area – usually a rather ample and planar space where load units are positioned directly on the ground and stacked vertically (such in the case of containers) or queued horizontally (such as in the case of swap bodies and pallets) – and of the handled load units – usually large and heavy items requiring special handling equipment.

Yard storage areas are commonly linked to sea or land ports but can also be found in large factories such as steel plants or tile factories. In this context, the processes involved are generally represented by i) the stowing of incoming load units (which implies the allocation of the space in the yard), ii) the picking of outgoing load units, and iii) the repositioning of load units preventing the picking of other units (also referred to as *reshuffle* or *marshalling*) (Jovanovic & Voß, 2014).

An open yard can be a highly dynamic scenario in which there are continuous arrivals and departures of load units: therefore, due to the difficulties related to the handling of massive items, the correct allocation of incoming load units in the yards is of paramount importance since it will affect the performance of the following processes. However, the goal of the allocation strictly depends upon the management policies, and in particular:

- If the items are always picked according to a last-in-first-out (LIFO) policy, then the load units can always be picked up directly (i.e. without repositioning other units blocking the required unit). In this case, the allocation problem may be reduced to minimising the distances travelled to stow and picking up the load units optimising operations sequence.
- If the items are picked according to a random sequence, other units may block access to a required unit. In this case, the main goal of allocation becomes the minimisation of the future handling operations, in particular related to the reshuffles.

In this paper, we are concerned with the latter case: as part of a research project in a steel plant, we are interested in investigating the yard management policies and the supporting models that can help the factory's operators in reducing costs, emissions and time losses by optimising the allocation of units in the yard. Being able to choose which heuristics best suit one's own business situation (given the characteristics and improvement objectives) from a pre-set list rather than inventing new ones is a great advantage for open-yards managers in terms of time, cost and staff effort.

1.1 Research goal and structure

In literature, the reshuffling problem in a storage yard is often referred to as the Block Relocation Problem (BRP) and is mainly studied in relation to containers seaports. The BRP concerns the search for the sequence of containers movements that minimises the number of operations to pick all the load units according to a priority rank from a storage yard (Li et al., 2020). Different optimisation models have been formulated for this problem. However, these models are usually highly expensive in terms of solution times and resources, and consequently, they are not very practical for operational use (Jovanovic et al., 2019a). For this reason, numerous heuristics have been developed, each with different characteristics, levels of complexity, and effectiveness.

Due to these premises, this paper aims to review the main heuristics existing in the scientific literature to resolve BRPs. At the time of writing this paper, there are no reviews of heuristics on this topic in the literature. In particular, the goal of the analysis is to answer the following research questions:

- RQ1: Which are the main characteristics, strengths and weaknesses of the heuristics existing in literature for the BRPs?
- RQ2: Which are the main prerequisites to apply the heuristics for the BRP in a non-maritime open yard?

The results of a Systematic Literature Review (SLR) are presented to answer these questions.

The research is structured as follows. Section 2 presents the context analysis concerning the block relocation problem in its maritime and non-maritime applications and a brief digression of the heuristics role. Section 3 illustrates the SLR methodology used in this research. Section 4 describes the main results from the SLR answering the two research questions. Finally, section 5 reports the conclusions of this research work.

2. Context analysis

2.1 The Block Relocation Problem

Picking up cargo units from a yard in order of priority and with as few movements as possible is a problem that has been extensively studied in the scientific literature related to seaports, particularly in container yard management.
The BRP is formally defined as follows by Petering and Hussein (2013). Consider C containers numbered from 1 to C that are temporarily stored in a storage yard. As typical in a container yard, these containers are stacked directly on top of each other in a storage bay consisting of S last-in-first-out (LIFO) stacks: all new units are stored on top of the stack, and units can be removed only from the top of the stack. As the time to move these containers approaches, management learns that the containers must be retrieved from the bay according to the sequence 1, 2, 3, …, C (i.e. container 1 must be retrieved first, container 2 must be retrieved second, and so on until all containers have been picked). It is important to underline that containers in each stack are not necessarily sorted according to their order in the movements sequence; therefore, reshufflings are usually needed. Containers that have not been retrieved must remain in one of the bay's S stacks until their retrieval time arrives. The goal is to retrieve all C containers from the bay using the minimum number of moves. A move could be either a container direct retrieval (i.e., a container is permanently taken out of the bay) or a reshuffle (i.e., a container is moved from one stack to another stack to free a container beneath). Figure 1 is an example of a BRP configuration in a maritime yard in which C=6 and three stacks.
The formulation of this problem assumes the following prerequisites:

- the loading units are uniform and of the same size;
- there must be a maximum number of containers that can be stored in a stack (i.e. the maximum height of a stack);
- the dates and times of pick-up of each container are known;
- the precedence of pick-ups among blocks is known;
- the yard is entirely mapped: the exact location of each loading unit must be known.

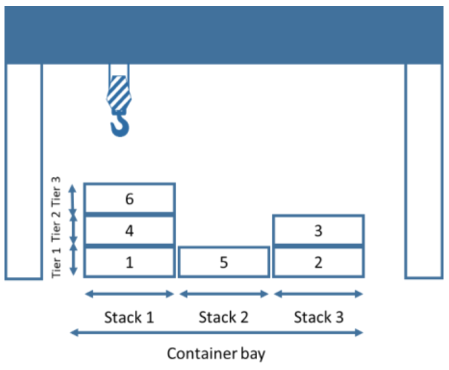

Figure 1: Example of a BRP configuration in a maritime yard (frontal view).

Furthermore, we assume that a load unit can only be moved when no other units are above it, and reshuffle occurs only at the moment when a target load unit is to be picked up. Finally, load units are reshuffled to other stacks in the same yard.
Even if this problem is mainly studied concerning seaport contexts, it is possible to apply it with similar assumptions and requirements in a factory open yard where the same elements are present. Indeed, the main difference is that while in the case of containers, the stacks are "vertically oriented" (containers are stacked on top of each other), in the case of a steel plant, the stacks are "horizontally oriented" (units are queued one after the other). In both cases, the stacks operate according to a LIFO policy. Figure 2 reports the same example shown in Figure 1 but in a horizontal open yard configuration.

The BRPs are NP-hard problems, as demonstrated by Caserta et al. (2012), and consequently, they are challenging to solve with an exact method (i.e., finding an optimal solution) in a real and operational context in which solutions have to be found in short or real-time. For this reason, a large number of heuristics for solving BRPs have been developed over the years in the yard management context.

2.2 The role of heuristics in the BRP

A heuristic or a heuristic method is defined as "a procedure for solving a well-defined (mathematical) problem by an intuitive approach in which the structure of the problem can be interpreted and exploited intelligently to obtain a reasonable solution" (Silver et al., 1980). In particular, heuristics are extremely useful in the following cases: i) when, although an exact analytic or iterative solution procedure exists, it may be computationally prohibitive to use or unrealistic in its data requirements; ii) as part of an iterative procedure that guarantees an optimal solution to quickly obtain an initial feasible solution or decide at an intermediate step of an exact solution procedure.

In summary, heuristics are mainly used when exact problem-solving techniques are too complex or take too long to solve with respect to the problem objective. For example, applying a BRP problem in an open-yard environment, the problem has to be solved every time a load unit or a group of units has to be picked up from the yard. Consequently, to be useful for operational purposes, the time to solve the problem must be proportional to the time taken to pick up a box.

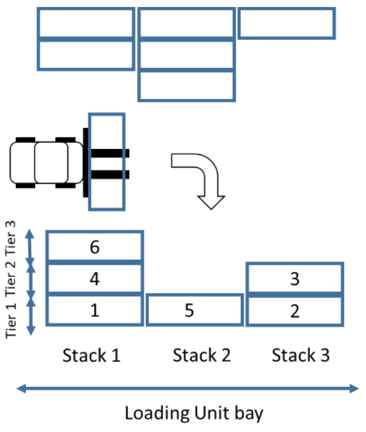

Figure 2: Example of a BRP configuration in an open yard (view from above)

3. Methodology

This paper aims to review the most commonly used heuristics in the literature for solving the BRP. In particular, this paper aims to find an answer to the two research questions presented in Section 1 through a Systematic Literature Review (SLR). A three-step protocol was developed to identify a proper procedure for performing automated research to guarantee the reliability of the SLR (Lagorio et al., 2020), as discussed in the following subsections.

3.1 Step 1: Inclusion/exclusion criteria

Firstly, the authors established the inclusion criteria for the search. A keywords research string was defined to limit the search field as much as possible, given the circumscribed problem to be explored. The review was limited to peer-reviewed publications to gain consistency between themes and sources and ensure the selected papers quality (Touboulic & Walker, 2015). The time interval considered was from 2000 to 2020, starting from the first mention in the literature of a heuristic for BRP resolution up to the present day. The search was launched based on the set of criteria reported in Table 1. Major publishers' databases and library services (i.e., Scopus and Web of Science), were selected for the analysis. This research led to the extraction of 19 papers.

Table 1: Inclusion criteria

Inclusion Criteria	Description
Keywords research string	"block* relocation problem*" AND "heuristic*"
Language	English
Document types	Articles
Source types	Peer-reviewed journals
Time Interval	2000 - 2020

3.2 Step 2: Selection based on title and abstract

Each author reviewed the titles and abstracts of selected papers. Following a discussion among the authors, papers out of the research scope were removed from the corpus. In particular, only one paper was excluded because it strongly related to an intermodal yard (Ji et al., 2016).

3.3 Step 3: Snowballing and final selection

The authors read the full versions of the candidate papers, and then no other paper was excluded. At this point, a corpus of 18 papers had been analysed. After that, a forward and backward snowballing process was conducted (Wohlin, 2014). Backward snowballing exploits the reference list to identify potential new papers to be included. The authors read titles, abstracts and full papers if necessary and then decided whether to include them in the final corpus. Forward snowballing identifies new papers starting from the analysis of papers that cited the ones contained in the first list of 18 papers. The two procedures were iterated until no new papers were found, and the final list includes 33 papers.

4. Discussion

The authors identified 43 heuristics described in the corpus of 33 papers (as some papers contained more than one heuristic). The authors read the papers by inductively searching for the main characteristics according to which the heuristics were classified. These characteristics, reported in Appendix A at the end of the paper, are respectively i) the arrival of the load units (static or dynamic), ii) the cleaning moves related to the withdrawal of the load units (restricted or unrestricted), iii) the type of heuristic (fast or slow), iv) the general characteristics of the heuristics. Appendix A also shows the exact method the heuristics refer to. As mentioned in Section 2.2, some heuristics are used to "simplify" some complex methods, e.g. by relaxing their constraints, excluding some elements or reducing the search range of the solutions. In these cases, the heuristics arise from exact models and produce a simplified version that does not lead to the problem optimal solution but provides a good approximation. The following subsections will discuss all these characteristics, thereby answering the research questions. The following features provide a first basis for understanding how to implement the most correct and efficient heuristics in real cases.

4.1 Load units arrival

The arrival of new load units to be stored in the yard could be forbidden (leading to a *static problem* where the goal is generally to empty the yard) or allowed (thus leading to a *dynamic problem*). In the corpus considered in this study, 38 heuristics deal with static BRPs, four heuristics examine dynamic BRPs, while only heuristics consider both static and dynamic versions of the problem. Dynamic problems entail higher complexity since the goal is not limited to empty the storage but is protracted over time to guarantee the minimisation of reshuffles movements. In dynamic cases, indeed, it may happen that a load unit entering the yard storage at time *t* will be picked up before other units already in the storage ad time *t*.

4.2 Cleaning moves

Cleaning moves are meant to relocate load units in the yard. The problem is termed *restricted* if only the load units above the unit that should be picked (referred to as the target unit) can be relocated. Otherwise, in an *unrestricted* problem, pre-marshalling moves are allowed, meaning that load units in stacks not containing the target units can be relocated to "cleaning" the storage and sorting the units according to their priority. Unrestricted problems imply the necessity for look-ahead approaches, estimating the possible future configurations of the yard. Although this is possible for static problems, it may be quite tricky in dynamic problems. In the considered sample, 32 heuristics involve restricted BRPs, 5 heuristics involve unrestricted BRPs, while 6 heuristics consider both restricted and unrestricted versions of the problem.

4.3 Types of heuristics

According to (Bacci et al., 2018), we distinguish between *fast* and *slow* heuristics. Fast heuristics are mainly based on greedy approaches, easy to apply (requiring limited amounts of information), and generally "original" in the sense that they do not originate from an exact model of problem-solving. On the contrary, slow heuristics are mainly based on optimisation models, are complex, require longer resolution times than fast heuristics, and often require a large amount of real data to guarantee solutions comparable to the optimal ones obtained by applying exhaustive models. In the considered sample, 19 heuristics are fast, and 24 are slow.

4.4 Main strength and weaknesses

Analysing the heuristics in the sample in detail, it emerged that there are no peculiar advantages and disadvantages to each heuristic, but in general, the heuristics presented have the same advantages and disadvantages common to all heuristics. On the one hand, the heuristics in the sample allow complex problems to be solved in a shorter calculation time, guaranteeing good (or at least acceptable) solutions comparable with the optimal ones obtained with the exact methods. On the other hand, the heuristics presented suffer from a common problem: when the problem size increases to very large instances, the heuristic solution gets further and further away from the optimal solution providing worse performance. In the years, thanks also to the fact that most of the papers analysed appeared in journals related to operations research, and therefore often reported with great care the steps taken for the construction and application of the heuristic, if not even the code, it was possible for researchers who came later to compare the performance of the new heuristics with the existing ones, sometimes even using the same database of data in origin. This good practice made it possible to have heuristics that continually perform better than their predecessors, guaranteeing better solutions even for larger instances.

4.5 Characteristics for application in non-maritime open-yards

All analysed heuristics are applied in the management of containers in seaport yards. However, many authors claim that with a few relatively simple modifications, the same heuristics can also be applied in non-port areas. Excluding the model proposed by Li et al., 2020, which also takes the vessel part into account, the others are all applicable to the non-maritime context. The only variables often found in heuristics that explicitly refer to the port context are those related to the manoeuvring and moving times of port cranes. However, as these are temporal variables, it is not difficult to consider data on forklift movement times instead of crane movement times.

5. Conclusions

This research represents a step in a broader research process aiming to identify yard management policies and supporting models that can be applied in other open yards, such as in steel plants. With respect to RQs, the main characteristics, strengths and weaknesses of heuristics (RQ1) are discussed from Section 4.1 to 4.4. Concerning RQ2, we did not find specific requirements for applying the heuristics to non-maritime yards other than those discussed in Section 2.1. Due to the constrained space available, this paper is limited to summarising state-of-the-art about the block relocation problem and its applications. However, such a step is important to drive the future analysis of the heuristics and their performance in general cases other that the seaport context providing useful support to open yard managers.

References

Akyüz, M. H., & Lee, C.-Y. (2014). A mathematical formulation and efficient heuristics for the dynamic container relocation problem. *Naval Research Logistics (NRL)*, *61*(2), 101–118.

Bacci, T., Mattia, S., & Ventura, P. (2018). *A new heuristic algorithm for the restricted Block Relocation Problem*. 32.

Bacci, T., Mattia, S., & Ventura, P. (2019). The bounded beam search algorithm for the block relocation problem. *Computers & Operations Research*, *103*, 252–264.

Caserta, M., Schwarze, S., & Voß, S. (2012). A mathematical formulation and complexity considerations for the blocks relocation problem. *European Journal of Operational Research*, *219*(1), 96–104.

Caserta, M., Schwarze, S., & Voß, S. (2009). A New Binary Description of the Blocks Relocation Problem and Benefits in a Look Ahead Heuristic. *Proceedings of the 9th European Conference on Evolutionary Computation in Combinatorial Optimization*, 37–48.

Caserta, M., Voß, S., & Sniedovich, M. (2011). Applying the corridor method to a blocks relocation problem. *OR Spectrum*, *33*(4), 915–929.

Casey, B., & Kozan, E. (2012). Optimising container storage processes at multimodal terminals. *Journal of the Operational Research Society*, *63*(8), 1126–1142.

Expósito-Izquierdo, C., Melián-Batista, B., & Marcos Moreno-Vega, J. (2014). A domain-specific knowledge-based heuristic for the Blocks Relocation Problem. *Advanced Engineering Informatics*, *28*(4), 327–343.

Feillet, D., Parragh, S. N., & Tricoire, F. (2019). A local-search based heuristic for the unrestricted block relocation problem. *Computers & Operations Research*, *108*, 44–56.

Forster, F., & Bortfeldt, A. (2012). A tree search procedure for the container relocation problem. *Computers & Operations Research*, *39*(2), 299–309.

Galle, V., Manshadi, V. H., Boroujeni, S. B., Barnhart, C., & Jaillet, P. (2018). The Stochastic Container Relocation Problem. *Transportation Science*, *52*(5), 1035–1058.

Jin, B., Zhu, W., & Lim, A. (2015). Solving the container relocation problem by an improved greedy look-ahead heuristic. *European Journal of Operational Research*, *240*(3), 837–847.

Jovanovic, R., Tanaka, S., Nishi, T., & Voß, S. (2019a). A GRASP approach for solving the Blocks Relocation Problem with Stowage Plan. *Flexible Services and Manufacturing Journal*, *31*(3), 702–729.

Jovanovic, R., Tuba, M., & Voß, S. (2019b). An efficient ant colony optimization algorithm for the blocks relocation problem. *European Journal of Operational Research*, *274*(1), 78–90.

Jovanovic, R., & Voß, S. (2014). A chain heuristic for the Blocks Relocation Problem. *Computers & Industrial Engineering*, *75*, 79–86.

Kim, K. H., & Hong, G.-P. (2006). A heuristic rule for relocating blocks. *Computers & Operations Research*, *33*(4), 940–954.

Ku, D., & Arthanari, T. S. (2016). On the abstraction method for the container relocation problem. *Computers and Operations Research*, *68*(C), 110–122.

Lagorio, A., Zenezini, G., Mangano, G., & Pinto, R. (2020). A systematic literature review of innovative technologies adopted in logistics management. *International Journal of Logistics Research and Applications*, *0*(0), 1–24.

Lee, Y., & Lee, Y.-J. (2010). A heuristic for retrieving containers from a yard. *Computers and Operations Research*, *37*(6), 1139–1147.

Li, J., Zhang, Y., Liu, Z., & Liang, X. (2020). Optimizing the Stowage Planning and Block Relocation Problem in Inland Container Shipping. *IEEE Access*, *8*, 207499–207514.

López-Plata, I., Expósito-Izquierdo, C., Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2017). Minimizing the Waiting Times of block retrieval operations in stacking facilities. *Computers & Industrial Engineering*, *103*, 70–84.

López-Plata, I., Expósito-Izquierdo, C., & Moreno-Vega, J. M. (2019). Minimizing the operating cost of block retrieval operations in stacking facilities. *Computers & Industrial Engineering*, *136*, 436–452.

Lu, C., Zeng, B., & Liu, S. (2020). A Study on the Block Relocation Problem: Lower Bound Derivations and Strong Formulations. *IEEE Transactions on Automation Science and Engineering*, *17*(4), 1829–1853.

Murty, K. G., Wan, Y., Liu, J., Tseng, M. M., Leung, E., Lai, K., & Chiu, H. W. C. (2005). *Hongkong International Terminals Gains Elastic Capacity Using a Data-Intensive Decision-Support System.*

Petering, M. E. H., & Hussein, M. I. (2013). A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem. *European Journal of Operational Research*, *231*(1), 120–130.

Silver, E. A., Victor, R., Vidal, V., & de Werra, D. (1980). A tutorial on heuristic methods. *European Journal of Operational Research*, *5*(3), 153–162.

Touboulic, A., & Walker, H. (2015). Theories in sustainable supply chain management: A structured literature review. *International Journal of Physical Distribution & Logistics Management*, *45*(1/2), 16–42.

Tricoire, F., Scagnetti, J., & Beham, A. (2018). New insights on the block relocation problem. *Computers & Operations Research*, *89*, 127–139.

Ünlüyurt, T., & Aydın, C. (2012). Improved rehandling strategies for the container retrieval process. *Journal of Advanced Transportation*, *46*(4), 378–393.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 1–10.

Wu, K., & Ting, C.-J. (2010). A beam search algorithm for minimizing reshuffle operations at container yards. In *Proceedings of the international conference on logistics and maritime systems,* 9, 15-17.

Wu, K. C., Ting, C. J., & HERNÁNDEZ, R. (2010). Appling tabu search for minimizing reshuffle operations at container yards. Journal of the Eastern Asia Society for Transportation Studies, 8, 2379-2393.

Zehendner, E., & Feillet, D. (2012). Column Generation for the Container Relocation Problem. *Progress in Material Handling Research: 2012.*

Zhu, W., Qin, H., Lim, A., & Zhang, H. (2012). Iterative Deepening A* Algorithms for the Container Relocation Problem. *IEEE Transactions on Automation Science and Engineering*, *9*(4), 710–722.

Appendix A: Characteristics of the heuristics

	Heuristics	Exact methods	B	C	T	General information	Reference
1	Lowest-slot (LS) heuristic	-	S	R	F	Simple and optimal for FIFO strategies	Zhang, 2000;
2	Reshuffle Index (RI) heuristic	-	S	R	F	It considers both the number of containers in a stack and their retrieval times	Murty et al., 2005
3	Expected Number of Additional Relocations heuristic (ENAR)	-	S	R	F	It considers the expected number of future reshuffles, the number of reshuffles is not sensible to the number of tiers and stacks, there are no tie situation	Kim & Hong, 2006
4	Extended LS, RI, ENAR heuristics	-	S	R	F	Same as the original LS, RI, ENAR heuristics	Wan et al., 2009
5	Minimum Reshuffle Integer Program (MRIPk) heuristic	IP	S/D	R	S	It does not consider the retrieval of all containers, but solve k subproblems	Wan et al., 2009
6	Matrix-Algorithm (MA) heuristic	-	S	R	S	Short computational time even for large problems	Caserta et al., 2009
7	Reshuffle Index with Look-Ahead (RIL) heuristic	-	S	R	F	Same of the RI with a rule for breaking ties: the look-ahead rule selects a stack in which the highest priority of the containers is the lowest, preventing future reshuffles	K. Wu & Ting, 2010
8	Three phase-heuristic (3PH)	-	S	R	S	It considers reduction both in terms of movements and working times	Lee & Lee, 2010
9	Tabu Search (TS) heuristic	B&B search	S	R	S	Good solutions even on large instances	K.-C. Wu et al., 2010
10	Beam Search (S) heuristic	-	S	R	S	Optimal solutions for small size instances	K. Wu & Ting, 2010
11	Corridor Method (CM) heuristic	DP	S	R	S	Optimal solutions even in large instances	Caserta et al., 2011
12	Min-max heuristic	-	S	R/U	F	It improves the ENAR heuristics with good solutions in a short time, even for large instances	Caserta et al., 2012
13	Lowest Absolute Difference (LAD) heuristic	-	S	R	F	It improves LI heuristic	Wu & Ting, 2012;
14	Group Assignment Heuristic (GAH)	-	S	R	F	It relocates a group of blocks instead of only one	Wu & Ting, 2012
15	Greedy heuristic	-	S	R	F	It minimises the total time of retrieval choosing the minimum possible value among all the expected additional relocations calculated for each stack	Ünlüyurt & Aydın, 2012
16	Difference heuristic	-	S	R	F	It minimises the difference in the orders of containers to minimise the number of containers that would potentially create recent relocations in the future	Ünlüyurt & Aydın, 2012
17	Probe heuristic	-	S	R/U	F	It leverages the advantages of the LS, RI and min-max heuristic considering both restricted and unrestricted variants	Zhu et al., 2012
18	Iterative deepening A* (IDA*) heuristic	IDA* search	S	R/U	S	It combines the branch and bound methods with two heuristics to determine the lower and upper (Probe heuristics) bounds	Zhu et al., 2012
19	Constructive heuristics with meta-heuristics	IP	D	R	S	It produces solutions with zero reshuffle moves for small-medium instances	Casey & Kozan, 2012
20	Heuristics column generation	IP	S	R	S	Optimal results for small and medium instances	Zehendner & Feillet, 2012
21	Heuristic tree search	-	S	R	S	It proposes several fixed classification schemes for reshuffles	Forster & Bortfeldt, 2012
22	Look-ahead heuristics algorithm (LA)	-	S	R	F	The heuristics have been tested compared to the previous slow ones, and it is better in terms of performance (CPU time and number of movements)	Petering & Hussein, 2013
23	Look-ahead (LA-N) heuristic algorithm	-	S	U	F	The heuristics have been tested compared to the previous slow ones, and it is better in terms of performance (CPU time and number of movements)	Petering & Hussein, 2013

24	Chain heuristic	-	S	R	F	It relocates a group of blocks instead of only one giving an improved version of the min-max algorithm and it is good only when the number of containers increases	Jovanovic & Voß, 2014
25	Height Index-Based (HIB) heuristic	-	D	R	F	Enhancements of the existing RI heuristic	Akyüz & Lee, 2014
26	Average Time Index-Based (ATIB) heuristic	-	D	R	F	Enhancements of the existing RI heuristic considering the average retrieving time for each stack	Akyüz & Lee, 2014
27	A Beam Search heuristic (BS)	IP	D	R	S	It is useful for the pre-marshalling improving aversion of the (S) algorithm	Akyüz & Lee, 2014
28	Domain-specific heuristics	A* search	S	R/U	S	It is based on a set of three easy rules for the reshuffling	Expósito-Izquierdo et al., 2014
29	Greedy look-ahead search (GLAS)	-	S	U	S	It is an improved version of the look-ahead heuristic	Jin et al., 2015
30	Depth-first branch & bound (DFBnB) heuristic	B&B search	S	R	S	Optimal solution for small-medium problems	Ku & Arthanari, 2016
31	Look-Ahead strategy considering waiting times	IP	S	R	S	It considers waiting time as a measure of the quality of service, it works on group of blocks improving the LA strategy	López-Plata et al., 2017
32	Rake search	B&B search	S	U	S	It isbetter than other heuristics for larger instances	Tricoire et al., 2018
33	Expected MinMax (EM)	-	S	R	F	It is based on min-max heuristic and group assignment heuristic and it considers probabilities of reshuffling	Galle et al., 2018
34	Expected Group Assignment (EG)	-	S	R	F	It is based on min-max heuristic and group assignment heuristic and it considers probabilities of reshuffling	Galle et al., 2018
35	Batch model	PBFS	S	R	S	It considers no "full information" hypothesis: the complete retrieval order at the beginning of the retrieval process is unknown. It considers a batch of containers and waiting times	Galle et al., 2018
36	Bounded Beam Search Algorithm	-	S	R	S	It improves previous similar heuristics	Bacci et al., 2019
37	Extended Greedy algorithm	-	S	R/U	F	The reshuffle of well-located blocks will be allowed in some specific case	Jovanovic, Tuba, et al., 2019
38	ant colony optimisation (ACO) algorithm	Ant Colony Algorithm	S	R/U	S	It improves previous similar heuristics	Jovanovic, Tuba, et al., 2019
39	Local-search based heuristic	DP	S	U	S	It improves any heuristically generated solution optimising the sequence of unproductive moves	Feillet et al., 2019
40	Greedy randomised adaptive search procedure (GRASP)	-	S	R	S	It considers correction procedure to remove the moves having undesirable properties	Jovanovic, Tanaka, et al., 2019
41	A* heuristic	IP	S	R	S	It considers operating costs	López-Plata et al., 2019
42	Hybrid neighbourhood heuristic	IP	S	R	S	It works only for maritime instances	Li et al., 2020
43	Mixed integer programming (MIP) formulation	IP	S	U	S	It is highly customisable	Lu et al., 2020

Abbreviations Legenda: IP = integer programming; B&B search = Branch and Bound search; DP = Dynamic Programming; PBFS = Pruning-Best-First-Search; B = Block arrival (S = Static, D = Dynamic); C = Cleaning Moves (R = Restricted, U = Unrestricted); T = Typology (F = Fast, S= Slow).