CIRRELT
1971 – 2021

**CIRRELT-2021-14**

# Multi-period Bin Packing Model and Effective Constructive Heuristics for Corridor-based Logistics Capacity Planning

**Teodor Gabriel Crainic**
**Franklin Djeumou Fomeni**
**Walter Rei**

**March 2021**

**Bureau de Montréal**
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

**Bureau de Québec**
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

# Multi-period Bin Packing Model and Effective Constructive Heuristics for Corridor-based Logistics Capacity Planning[ł]

**Teodor Gabriel Crainic[*], Franklin Djeumou Fomeni, Walter Rei**

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

**Abstract.** The bin packing problem is one of the most studied combinatorial optimization problem. This paper proposes two novel bin packing problem settings with many practical applications, in particular in logistics capacity planning. Both problems explicitly consider, besides the classical bin-selection costs, the item and bin-specific item-to-bin assignment costs. These assignment costs depend not only on the physical, e.g., item and bin size, and economic, e.g., bin selection fixed cost and the cost of item \transport" by the bin, but also, on the temporal attributes of items and bins, e.g., availability of regular bins for selection and utilization and of items to be assigned to such a regular bin. Special, item-specific in terms of size, spot-market bins may be used at higher cost for the items one cannot fit into the selected bins. Single and a multi-period formulation are proposed, both aiming to minimize the total cost of the system computed as the sum of the fixed costs of the selected bins and the total item-to-bin assignment cost using regular and spot-market bins. The multi-period formulation optimizes the cost over all the time periods considered. Several constructive heuristics are proposed, three for the single-period model, and four for the multi-period formulation. The heuristics are evaluated and compared through an extensive computational experimentation. The numerical results show the high level of performance of the proposed heuristics in terms of solution quality and computational efficiency, as well as the potential benefits of using the new models in practical applications.

**Keywords**: Multi-period variable size and cost bin packing, item-to-bin assignment costs, constructive heuristics, logistics capacity planning, consolidation, integer programming.

**Acknowledgements.** While working on the project, the first author was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréal, while the second was Postdoctoral Researcher, School of Management, UQAM, and CIRRELT. We gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery, Acceleration Grant and Engage programs. We also gratefully acknowledge the support of Fonds de recherche Nature et Technologies du Québec (FRQNT)through their team research project program. Thanks are also due to Calcul Quebec and Compute Canada for providing the authors access to their high-performance computing infrastructure.

[ł]Revised version of the CIRRELT-2019-24



---

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

# 1 Introduction

Producers, wholesalers, and retailers are regularly faced with the need of moving goods across their distribution networks. This could be the need to ship goods from producers, suppliers, or warehouses to customers or outlets for stock replenishment. The costs involved in such movements of goods represent a significant part of the distribution and operating costs of these companies Monczka et al. (2008). Therefore, researchers have developed decision support tools, based on the use of bin packing models, to reduce such distribution costs Perboli et al. (2014).

Given a set of items and a set of bins, the classical *Bin Packing Problem (BPP)* aims to load all the items into the smallest possible number of bins, while ensuring that the packing of each bin is feasible Martello and Toth (1990); Dyckhoff (1990). Various applications of the BPP can be encountered in the planning of telecommunication, transportation, production and logistics/supply chain systems Crainic et al. (2011). In transportation and logistics system planning, the items may represent the goods that need to be moved or stored, while the bins represent the transportation medium, e.g., containers, container slots within ships or on trains, and capacitated vehicles of any given mode, or the storage means available. The classical BPP assumes that all the bins are identical in cost and size, thus focusing on the physical characteristics of the problem, e.g., the packing constraints, while ignoring several important considerations such as, the variability in the bin types, costs and sizes, the economic characteristics of item-to-bin assignment, and the temporal characteristics of when items and bins become available and decisions are to be taken Crainic et al. (2016).

The BPP has attracted a lot of attention over the past few decades. Several BPP variants have been studied in the literature, the BPPs with variable cost and size bins targeting the first concern above. Yet, as a review of the literature shows (Section 2), it still appears that the existing BPP variants focus on selecting the bins at the lowest cost possible and loading them with any items available. Hence, all these variants assume that the cost of loading an item into a bin depends on the item volume only. This is a serious limitation, as the cost of assigning an item to a bin may be influenced by many other considerations related to the item, the bin, and other decision-making characteristics. We discuss these issues in more depth in Section 3 but, for illustration purposes, consider the logistics case when loads (items) have to be delivered by a certain due date, and one must chose among several carrier services (bins) operating particular vehicles according to specific timetables. Bin packing offers the methodology to address this problem setting, but the current BPP variants, which focus on the bin cost only, are not appropriate.

The first main contributions of this paper is to bridge this gap in the literature and propose a new BPP variant, which addresses the issues identified above and extends the capabilities of the Bin Packing methodology. More precisely, we introduce a time-dependent BPP setting wherein the cost of assigning an item to a bin is explicitly ac-

counted for. Furthermore, the item-to-bin assignment cost may depended not only on the physical and cost attributes of items and bins, but also on the temporal characteristics of both and on when the item is assigned to the bin. We designate this version of BPP, the *Multi-period Variable Cost and Size Bin Packing Problem with Assignment Costs*. A static, *single-period* version of this problem setting explicitly considers the item-to-bin assignment costs, hence also generalizing previous models in the literature, but without the time component. We study this *Single-period Variable Cost and Size Bin Packing Problem with Assignment Costs* as well in this paper.

For efficient decision support when planning, e.g., logistics and transportation activities, bin-packing methods need to produce good-quality solutions rapidly. Consequently, in addition to the description and mathematical formulation of these two new BPPs, we also propose a set of constructive heuristic algorithms to address them; three for the single-period and four for the multi-period problem settings. The main idea behind these heuristics is to adapt the well-known *First-Fit Decreasing* and *Best-Fit Decreasing* heuristics (Martello and Toth, 1990; Dyckhoff, 1990) to the problems at hand. As computationally shown in Section 7, the proposed heuristic algorithms achieve these goals of rapidly providing high-quality solutions. They are therefore appropriate both for practical applications and as components of more sophisticated solution methods.

In order to properly assess the efficiency of the proposed heuristics, we carry out a thorough computational experiment with a set of 1540 instances of the single and multi-period problems. The objective of these experiments are fourfold. First, we evaluate the efficiency of the proposed heuristic algorithms. This efficiency is measured in terms of CPU computation time and solution quality, i.e., the optimality gap with respect to the exact solution provided by a well-known commercial solver, or the lower bound when the solver cannot reach the optimum within a given time limit. Second, we examine the impact on solutions of certain problem characteristics, in particular those related to the availability periods of items and bins. Third, we measure the impact of explicitly integrating the time-dependency of items, bins, and their related decisions into the bin packing formulations. We preform this analysis by evaluating the cost reductions provided by the multi-period formulation compared to a myopic approach addressing the same instances without considering the multi-period setting. Finally, we assess the ability of the solutions provided by the heuristics to improve the performance of commercial solvers. More precisely, we feed the objective-function value of the best heuristic solution to the solver as an initial upper bound for the branch-and-cut algorithm, which proves to increase significantly the node-pruning capability of the solver.

The results of the computational experiments confirm the interest of the new problem settings and models, the importance of explicitly consider the time esmension of the problem (one observes reductions of up to some 82% in operational costs provided by the multi-period formulation compared to a myopic approach), as well as the efficiency of the proposed heuristics. Indeed, while some problem instances with particular structures

can be solved by commercial solvers within a time limit of 1 hour, the solvers struggle to solve a large number of instances. The proposed constructive heuristics obtain good-quality solutions very rapidly, providing great trade-offs between computational time and optimality gap regardless of the structure of the problem at hand. One may thus use them either directly to address problem in practice or to provide efficient initial cut-off upper bounds that enhance the performance of commercial solvers.

The rest of this paper is structured as follows. We present a brief review of the literature related to the problem at hand in Section 2. Section 3 provides an in-depth description of the proposed novel variants of the BPP. The corresponding mathematical models are then presented in Section 4. Sections 5 and 6 describe the proposed heuristic algorithms for the single and multi-period models. The results and analyses of the computational experiments carried out are discussed in Section 7. Finally, Section 8 provides concluding remarks and insights for potential extensions of this work.

## 2 Literature Review

The problem studied in this paper generalizes previously-studied BPPs with varying characteristics, in particular the cost and capacity of the bins. Prior to briefly discussing the related literature, we introduce a labeling scheme to simplify the identification of the bin packing deterministic problem variants. The proposed notation is inspired by labeling schemes used in other fields, e.g., scheduling, queuing, and location-routing, as well as by previous attempts in the packing and cutting surveys, e.g., Dyckhoff (1990); Wäscher et al. (2007). We aim to enrich the current classifications and identify in a concise form the main cost, size, and time attributes and relations defining packing problem classes. We thus propose the labeling system $D/C/B/K/T[\cdot]$, where

- $D$ = Number of (physical, handling, etc.) dimensions; D = 1, 2, 3, ... ;
- $C$ = Bin (fixed) cost; C = **U**, unique and same for all bins, or **V**, variable and possibly different for each bin;
- $B$ = Bin size: B = **U**, unique (same for all bins) or equal/proportional to the cost of the bin, or **V**, variable and possibly different for each bin;
- $K$ = Commodities, that is, the number of different item types; K = **S**, single commodity / item type, or **M**, multicommodity / item types;
- $T$ = Time representation; T = **1**, single period, **N**, multi-period, and **C**, continuous; In this classical definition, single period means time is not explicitly represented and there are no time-related attributes for bins and items, all events and decisions taking place simultaneously; The decision-planing horizon is discretized into a number

of time periods in **N** problem settings, all time-related events (e.g., the starting of a bin availability interval) being associated to a particular time period; Finally, an event may happen at any time moment during the length of the decision-planning horizon in a **C** problem setting;

- [·] = List of additional attributes, such as item-to-bin assignment costs, *I2B* in the following, bin loading rules, permitted item rotation, inclusion/exclusion rules among items and bins, etc.

According to this labeling scheme, the problems studied in this paper are the 1/V/V/S/N[I2B] and 1/V/V/S/1[I2B], the Multi- and Single-Period Variable Cost and Size Bin Packing Problems with Assignment Costs, respectively.

It should be noted that the BPP and its variants belong to the broad field of general *packing* and *cutting* problems Dyckhoff (1990); Martello and Toth (1990); Dyckhoff et al. (1997); Wäscher et al. (2007); Brandão and Pedroso (2016); Crainic et al. (2012). The former aims to find how to best pack small objects into larger ones, while the latter focuses on how to best cut large objects into smaller items. While differences exist (see, e.g., Dyckhoff (1990)), the two are very much related and one can often reformulate one in the vocabulary and setting of the other one. The application context often indicates the most appropriate. In our case, inspired by logistics and transportation applications, the packing environment appears the most appropriate. Although a thorough literature survey of this vast field, and of related problem settings, e.g., *parallel machine scheduling* Lenstra et al. (1977); Fanjul-Peyro et al. (2017); Kravchenko and Werner (2011), *assembly* and *disassembly line balancing* Wee and Magazine (1982); Battaïa and Dolgui (2013); Álvarez-Miranda and Pereira (2019); Özceylan et al. (2019), and *irregular packing* Leao et al. (2020), is of interest to the research community, it is well beyond the scope of this paper focusing on a new problem setting in bin packing. We therefore now briefly discuss previous contributions directly relevant to the problems studied.

The class of *Variable Size Bin Packing Problems*, D/U/V/S/1, constitutes a first BPP setting related to the problem at hand Wäscher et al. (2007). This case considered bins of different sizes with costs either equal to their sizes or strictly proportional to those. The next step in addressing increasingly practical application concerns yielded the *Variable Cost and Size Bin Packing Problem* class, D/V/V/S/1, which disconnected the definition of the costs of the bins from their sizes Monaci (2002); Pisinger and Sigurd (2005); Kang and Park (2003); Alves and de Carvalho (2007); Crainic et al. (2011). A stochastic extension of the D/V/V/S/1, integrating uncertainty regarding future item and bin characteristics into current decisions on bin bookings, was proposed in Crainic et al. (2016), where it was applied to a logistics capacity planning problem.

The *Generalized Bin Packing Problem*, D/V/V/M=2/1[optional items], extends the D/V/V/S/1 by considering two categories of items, compulsory ones that have to be loaded into bins, and non-compulsory ones, which can either be loaded for an additional

profit or be left behind at no cost Baldi et al. (2012, 2014). Baldi et al. Baldi et al. (2019) extended the latter problem within the context of last-mile logistics by introducing bin-dependent profits earned by loading items into specific bins and maximizing the total net revenue. The authors called this setting the *Generalized Bin Packing Problem with Bin-dependent Item Profit*, D/V/V/M=2/1[optional items, bin-related item profits]. The proposed single-period 1/V/V/S/1[I2B] BPP and model adds new dimensions to the problem as highlighted in Section 4.3.

Bin packing problems with consideration of the time dimension have recently started to gain attention from the research community. Namely, Dell'Amico et al. Dell'Amico et al. (2020) recently introduced the temporal BPP (TBPP), as a generalisation of the standard BPP wherein each item has to be packed within a given time window. This variant considers identical bins, and may therefore be classified as 1/U/U/S/N[ITW] (Time Windows on Item assignment). An application of the TBPP in the context of cloud computing Aydın et al. (2020) resulted into a multi-objective formulation, for which a compact formulation that can be solved with commercial solvers is proposed in Martinovic et al. (2021).

This overview of the literature emphasizes that two major problem characteristics, often encountered in logistics and transportation applications (at least), are not currently addressed. The time-dependent availability of both items and bins of various types is ignored, as are the costs of assigning items to bins, which are not dependent only on the physical characteristics of both, but also and principally on the time of the assignment and the dispatching of the bin to deliver the items on time.

We conclude this section recalling that the BPP is known to be NP-hard in the strong sense Korte and Vygen (2006); Garey and Johnson (1979). Most BBP exact solution methods found in the literature rely either on decomposition techniques Alves and de Carvalho (2007); Monaci (2002); Casazza and Ceselli (2016) or on reformulations solved using commercial MIP software Correia et al. (2008). The computational effort required by these methods makes them difficult to use in practice, however, as they are limited to small size problems only Martello and Toth (1990). Consequently, heuristic solution methods are proposed and allow to find good solutions to larger size problems within reasonable computational efforts. The literature shows that the most commonly used constructive heuristics are the *First-Fit Decreasing* (FFD) and *Best-Fit Decreasing* (BFD) heuristics (Martello and Toth, 1990; Dyckhoff, 1990). Both heuristics first sort items in decreasing order of their size. The FFD heuristics then assigns each item, one after the other, to the first bin which can accommodate it, i.e., with sufficient residual capacity (nominal capacity minus the space occupied by the currently loaded items); a new bin is selected when no current bin can be used. BFT heuristics proceed in a similar way, except that each item is assigned to the best bin that can accommodate it. The contributions reviewed above are all implementing some form of these two constructive heuristics. We will extend this framework to develop the constructive heuristics we

propose for the single and multi-period BPP with variable bin cost and size and item-to-bin assignment costs.

# 3 Problem Description

The problem setting is inspired by the planning challenge of corridor-based integrated logistics systems supporting freight exchanges between two regions (e.g., South-West China and Canada). The system includes several shippers, e.g., producers, wholesalers, and retailers, which need to ship *items* from one region to the other. Inter-region transportation takes place between two major terminals, one in each region (e.g., the ports of Shenzhen and Vancouver, respectively), where the items to be shipped are brought in to be grouped for long-haul transport to the destination terminal, and from where incoming items are distributed within the region. The pickup and distribution activities are not explicitly considered in the problem setting, they are rather included in the assignment costs described bellow. Several transportation-service providers are also part of the system and offer a pool of transportation units, called *bins* in the following, among which the system selects the ones to move the items.

The planning problem is defined for a certain *schedule length* of finite duration (e.g., one week, one month), divided into *time periods* of equal length (e.g., one day). Given sets of items (the *demand*) and bins (the *supply*), with particular physical, temporal, and economic attributes, the goal is to find the selection of bins and the assignment of items to bins to satisfy the demand with the available supply at the minimum total cost over the entire schedule length.

Each item is characterized by its size, identified as *volume* in this single-dimension bin packing problem setting, as well as by particular handling and loading requirements (e.g., does it require refrigeration?), potentially related to particular transportation modes (e.g., rail or truck). Items are also characterized by temporal attributes, i.e., the times of 1) *availability*, when the item appears in the system and is available for shipping; 2) *delivery due* date, indicating the preferred moment the shipment should be at destination; 3) *earliest delivery*, when one can deliver earlier, possibly at a penalty cost; and 4) *latest delivery*, the time moment beyond which delivery is no longer acceptable, even at a penalty cost. Economic attributes are related to the costs associated to satisfying the demand for movement given the previous two classes of attributes, i.e., 1) *drayage* cost, related to the first/last-mile (kilometer) collection and delivery of the item within the respective origin and destination zones; 2) *dispatching-delay* cost, accounting for possible storage costs while waiting for assignment and departure; 3) *early* and *late-delivery* penalty costs when the item is delivered before or after its preferred due date, respectively. A mode-specific *transport* cost, related to the handling and loading requirements, links items and bins.

Indeed, bins are also characterized by physical, temporal, and economic attributes. A bin is thus characterized by its *volume*, or *capacity*, expressed in the same units as the item volumes, as well as by the *travel time* it requires to move between the two regions (terminals). Given the delivery restrictions and costs associated to items, it is important to account for the bin travel times when deciding on the item-to-bin assignments to enforce constraints at the lowest cost possible. The *availability interval* of a bin represents the time window within which it is available and can be used for loading and shipping items. The bin cannot be used once this interval has expired. The economic attributes of a bin are, first, its fixed selection cost and, second, the *transport* costs mentioned above, which are related to the handling and loading requirements of items as well as the bin's own mode and other operational characteristics. In the current problem setting, any bin can be loaded with any combination of items that fits into it.

In this setting, when a bin is made available to the system, one needs to decide not only whether to use (select) it or not, but also *when* to use it, that is, when, within its availability interval, to complete the loading and have it leave. Symmetrically, once an item is made available to the system, one needs to decide whether to load it into a bin that leaves immediately, or to delay and load it into a bin departing at a later time period, as long as it can be delivered before the latest delivery limit. When the need arises, e.g., when there is not sufficient bit capacity available for the items in need of shipping at that particular point in time, one may also call on non-participating transportation-service providers outside the pool at a, normally, higher unit item transportation cost. This ad-hoc decision is identified as using *spot-market* bins, in the following.

The overall objective of the problem is to minimize the total cost of the system for the schedule length, while satisfying the time and physical attributes of the given bins and items. The total system cost is computed as the sum of the *total fixed* cost of selecting the bins for usage at particular time periods within the schedule length, the *total variable assignment* (transportation) cost of items to the selected bins, and the *total variable ad-hoc* cost of assigning items to spot-market bins.

One should note that, differently from the literature on variable-cost-and-size bin packing (Section 2), the bins considered in the 1/V/V/S/N[I2B] may not only be selected and loaded once, but may be selected for use at a particular time period within a bin-specific interval. This decision, combined to the travel time of the bin, yields a particular time at destination, which has a direct impact on the feasibility, in terms of latest delivery, and cost of the assignment of items to be loaded into it. Thus, a feasible assignment of a particular item to a given bin implies a *time-dependent item-to-bin unit assignment cost*, which captures (through a direct sum, usually) the drayage cost, the transport cost, as well as any penalty for early or late delivery reflecting the contrast between the temporal attributes of the item and the decision to use the bin at a particular time period. It is noteworthy that the time-dependent item-to-bin assignment costs may be computed straightforwardly before the optimization starts.

The time dependency of items and bins, reflected in their temporal attributes, the bin selection decisions with restriction of using each bin at most once during their respective availability interval,s and the item-to-bin assignment costs dependent on the item and bin attributes make the 1/V/V/S/N[I2B] and the 1/V/V/S/1[I2B] stand apart from most problems currently present in the literature. They are, in fact, representative of a broad class of new BPP problems with weights ("costs") fully dependent on a multi-varied range of item and bin attributes.

It is noteworthy that the proposed models and algorithms for these deterministic problem settings, , described in the following sections, may be used in various contexts. On the one hand, the plan built for a particular schedule length given predicted item requests and bin availabilities may be used repeatedly over a medium-term tactical planning horizon. On the other hand, the 1/V/V/S/N[I2B] model and heuristics may be used in operations, its look-ahead capabilities strengthening the decisions of the current period.

We bring this section to a close noticing that, while the 1/V/V/S/N[I2B] problem setting represents a growing trend in collaborative or capacity-sharing logistics systems, including transportation and storage (e.g. Yilmaz and Savasaneril, 2012; Huang et al., 2013), the relevance of the bin packing methodology developed in this paper is broader. Consider, e.g., the case of hospital resource management during a major pandemic. Items would be patients, in this context, in need of hospitalization within certain temporal limits with various levels of dire consequences. Bins, on the other hand, would be beds available at different dedicated hospitals, which may be made available to infected patients when required, or be used for normal hospitalization, otherwise.

# 4 Mathematical models

We propose models for the multi-period 1/V/V/S/N[I2B] problem, described in Section 3), and for the single-period variant. The models are introduced in Sections 4.2 and 4.3, respectively, following the notation description.

## 4.1 Notation

Let $T$ designate the duration of the schedule length in terms of number of time periods. Let $\mathcal{I}$ be set of the items and $\mathcal{J}$ the set of bins which are made available during the schedule length. We define first the notation particular to items and bins, followed by the notation of the temporal and economic aspects of item-to-bin assignment.

For each item $i \in \mathcal{I}$, define

- $v_i$: Volume;
- $\underline{t}_i$: Availability time;
- $\bar{t}_i$: Latest delivery time, with $\mathcal{T}_i = [\underline{t}_i, \bar{t}_i]$;
- $\gamma_i$: Earliest delivery time;
- $\delta_i$: Delivery due date, with $\underline{t}_i \leq \gamma_i \leq \delta_i \leq \bar{t}_i$;
- $p_i^t$: Item-to-spot-market-bin assignment cost, i.e., the cost of shipping through the spot market at period $t \in \mathcal{T}_i$.

For each bin $j \in \mathcal{J}$, define

- $f_j$: Fixed cost of selecting and using the bin;
- $V_j$: Volume (capacity);
- $\Gamma_j = [\underline{\Gamma}_j, \overline{\Gamma}_j]$: Availability time window;
- $\alpha_j$: Travel time.

Let $\mathcal{T}_{ij}$ be the item-to-bin assignment interval, that is, the time window within which it is feasible to assign item $i \in \mathcal{I}$ to bin $j \in \mathcal{J}$, where feasibility is defined with respect to the availability interval of the item and bin, i.e. $\mathcal{T}_{ij} = [\underline{t}_i, \bar{t}_i - \alpha_j] \cap \Gamma_j$. Let, then, $a_{ij}^t$ be the corresponding *item-to-bin assignment cost* of item $i \in \mathcal{I}$ to bin $j \in \mathcal{J}$ at period $t \in \mathcal{T}_{ij}$ (recall that $a_{ij}^t$ captures the various costs as described in Section 3).

## 4.2 Formulation of the multi-period model

Define the following decision variables

- $y_j^t = \begin{cases} 1 & \text{if bin } j \in \mathcal{J} \text{ is selected at time period } t \in \Gamma_j, \\ 0 & \text{otherwise;} \end{cases}$
- $x_{ij}^t = \begin{cases} 1 & \text{if item } i \in \mathcal{I} \text{ is assigned to bin } j \in \mathcal{J} \text{ at time period } t \in \mathcal{T}_{ij}, \\ 0 & \text{otherwise;} \end{cases}$

- $u_i^t = \begin{cases} 1 & \text{if item } i \in \mathcal{I} \text{ is assigned to a spot-market bin at time period } t \in \mathcal{T}_i, \\ 0 & \text{otherwise.} \end{cases}$

The multi-period 1/V/V/S/N[I2B] model can formally be written as follows:

$$\min_{y,x,u} \quad \sum_{j\in\mathcal{J}} \sum_{t\in\Gamma_j} f_j y_j^t + \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}_{ij}} a_{ij}^t x_{ij}^t + \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{T}_i} p_i^t u_i^t \tag{1a}$$

$$s.t. \quad \sum_{i\in\mathcal{I}} v_i x_{ij}^t \leq V_j y_j^t, \qquad \forall j \in \mathcal{J}, t \in \Gamma_j, \tag{1b}$$

$$\sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}_{ij}} x_{ij}^t + \sum_{t\in\mathcal{T}_i} u_i^t = 1, \qquad \forall i \in \mathcal{I}, \tag{1c}$$

$$\sum_{t\in\Gamma_j} y_j^t \leq 1, \qquad \forall j \in \mathcal{J}, \tag{1d}$$

$$y_j^t \in \{0,1\}, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}_j, \tag{1e}$$

$$x_{ij}^t \in \{0,1\}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}_{ij}, \tag{1f}$$

$$u_i^t \in \{0,1\}, \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}_i. \tag{1g}$$

The three terms of the objective function (1a) are, respectively, the total cost of selecting the bins, the total cost of assigning the items to the bins at feasible time periods, and the total cost of using the spot market. Constraints (1b) ensure that the volume capacity of each bin cannot be exceeded. Constraints (1c) enforce satisfaction of demand, i.e., each item is assigned to a bin or to a spot-market bin within its availability time interval. Constraints (1d) ensure that each bin is selected at most once during its availability time window. Finally, constraints (1e)–(1g) enforce the binary requirements of the decision variables.

## 4.3 Formulation of the single-period model

The mathematical formulation of the single-period 1/V/V/S/1[I2B] model is similar to that of the general multi-period one (Section 4.2), except for the time index on parameters

and decision variables:

$$\min_{y,x,u} \quad \sum_{j\in\mathcal{J}} f_j y_j + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} a_{ij}x_{ij} + \sum_{i\in\mathcal{I}} p_i u_i \tag{2a}$$

$$s.t. \quad \sum_{i\in\mathcal{I}} v_i x_{ij} \leq V_j y_j, \qquad \forall j \in \mathcal{J}, \tag{2b}$$

$$\sum_{j\in\mathcal{J}} x_{ij} + u_i = 1, \qquad \forall i \in \mathcal{I}, \tag{2c}$$

$$y_j \in \{0,1\}, \qquad \forall j \in \mathcal{J}, \tag{2d}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{2e}$$

$$u_i \in \{0,1\}, \qquad \forall i \in \mathcal{I}, \tag{2f}$$

where the objective function (2a) minimizes the total cost of selecting the bins, assigning items to bins, and using the spot market; constraints (2b) enforce the bin capacity; constraints (2c) ensure that each item is either assigned to a bin or shipped via the spot market; constraints (2d)–(2f) enforce the binary requirements of the variables.

Note that one can find a feasible solution to the 1/V/V/S/N[I2B] by solving $T$ instances of the 1/V/V/S/1[I2B], one instance for each time period defined by the sets of items and bins becoming available at the corresponding time period, without considering future time periods. We explore this heuristic strategy in the computational experiments presented in Section 7 where it is used as benchmark in assessing the value of the proposed 1/V/V/S/N[I2B] methodology.

Recall that the 1/V/V/S/1[I2B] is more than a variant of the multi-period formulation. It is a problem and model on its own, generalizing the variable size and cost bin packing class of problems by explicitly considering, other than the fixed cost of selecting bins, item and bin attribute-based item-to-bin assignment costs. It is important to point out in this context that, by considering negative item-to-bin assignment costs, the single-period problem we propose is close to the BPP with optional items and bin-related item profits of (Baldi et al., 2019). It also generalizes it by, on the one hand, assuming each bin may be of a different type and, on the other hand, each item may be loaded into its own ad-hoc bin with capacity equal to the item volume and selection cost equal to the item-to-spot-market-bin assignment cost, instead of loading in to dummy bin of indistinct but very large capacity and cost. This generalization corresponds to a broader gamut of applications (e.g., sending shipments individually and paying the proper price) and offers a broader scope to the optimization of the resulting model.

# 5 Heuristic algorithms for the 1/V/V/S/1[I2B] model

We propose three constructive heuristic algorithms, *HS1 - HS3.* to efficiently find feasible solutions to the 1/V/V/S/1[I2B]. HS1 is item-centric, in the sense that it focuses on finding the best bin into which each item can be loaded. HS2 and HS3, on the other hand, are bin-centric focusing on finding the best packing for each bin by considering the possible contribution to the total cost of the items which could be loaded. The details on how this contribution is computed in each of these two heuristics are provided below.

## 5.1 Item-centric heuristic for the single-period problem

The main idea of the HS1 heuristic, Algorithm 1, is, first, to construct a feasible solution by finding for each item the most *suitable* bin in which it can be loaded, if such a bin has already been selected, or select a new suitable bin, otherwise. An item-swapping procedure and the usage of spot-market bins are called upon when the first phase does not yields a feasible solution (i.e., unassigned items still exist).

Given sets of already selected bins $\mathcal{S} \subset \mathcal{J}$ and items $\mathcal{A} \subset \mathcal{I}$ assigned to them, let $V_j^*$ be the residual capacity of bin $j \in \mathcal{S}$ and $\bar{a}_{ij} = a_{ij}/(V_j^* - v_i)$ the unit impact cost of using part of the residual capacity of $j \in \mathcal{S}$ for an unassigned item $i \in \mathcal{I}$.

The algorithm starts with empty sets of selected bins and assigned items. It then considers each unassigned item $i$ sequentially trying to load it into the most suitable bin among those already selected, where "suitable" is defined as the one with sufficient residual capacity, unit item-to-bin assignment cost less that the corresponding spot-market bin, and lowest unit impact cost. If such a bin exists, the item is assigned to it. Otherwise, a new bin is possibly selected, provided it is cost efficient to do so. The item is assigned to the new bin, if selected, or remains unselected, otherwise.

The evaluation and identification of a new bin to select is performed by Algorithm 2 implementing the $Profitable(i, b)$ procedure, which yields a new bin $b$ (if it exists) for unassigned item $i$. The idea is to measure the contribution of a new bin with respect to an estimation of its overall cost. Hence, a new bin is selected, and item $i$ is assigned to it, if there exists a subset of other unassigned items that can be assigned to the bin with an overall cost, fixed bin selection plus sum of the item-to-bin assignment costs, smaller than the sum of the individual item-to-spot-market-bin assignment costs.

If all items are assigned once all items have been investigated, the heuristic stops. Otherwise, the algorithm proceeds to evaluate possible swaps between unassigned items and assigned ones, aiming to fill up the residual bin capacities when it is profitable to do so. Any unassigned item the end is shipped via the spot market (i.e., assigned to a

spot-market bin).

**Algorithm 1** HS1

*Input:* Sets of items $\mathcal{I}$ and bins $\mathcal{J}$;
*Output:* Sets $\mathcal{A}$ of items assigned to bins and $\mathcal{S}$ of selected bins;
Initialize $\mathcal{S} = \emptyset$, $\mathcal{A} = \emptyset$;
**for** all $i \in \mathcal{I}$ **do**
  Identify the bin $b \in \mathcal{S}$ into which $i$ can be loaded at smallest values of $\bar{a}_{ij}$ and $a_{ij} < p_i$;
  **if** $b$ exists **then**
    Load $i$ into $b$;
    $\mathcal{A} := \mathcal{A} \cup \{i\}$;
  **else**
    Identify the bin $b \in \mathcal{J} \setminus \mathcal{S}$ using $Profitable(i, b)$;
    **if** $b$ exists **then**
      Load $i$ into $b$;
      $\mathcal{A} := \mathcal{A} \cup \{i\}$, $\mathcal{S} := \mathcal{S} \cup \{b\}$;
    **else**
      Leave $i$ as unassigned;
    **end if**
  **end if**
**end for**
# Look for possible swaps between assigned items and unassigned ones;
**for** $i \in \mathcal{A}$ assigned to a bin $j$ **do**
  **for** $i' \in \mathcal{I} \setminus \mathcal{A}$ **do**
    **if** $(V_j^* - v_{i'} + v_i \geq 0)$ and $(a_{ij} > a_{i'j})$ **then**
      Replace item $i$ with item $i'$ in bin $j$;
      Update the sets of assigned and unassigned items appropriately;
      $V_j^* := V_j^* - v_{i'} + v_i$;
    **end if**
  **end for**
**end for**
Assign any still unassigned item to a spot-market bin.

## 5.2 Bin-centric heuristics for the single-period problem

While the algorithm HS1 focuses of packing each item into the best possible bin and then complete the packing of the bin, the second and third heuristics aim to simultaneously select the bin and accompanying packing which makes the best contribution to the total cost.

Both heuristics sort the bins in increasing order of their *potential for contribution*

**Algorithm 2** $Profitable(i, b)$ procedure for new bin $b$ selection for an item $i$

*Input:* Sets of unassigned items $\mathcal{I} \setminus \mathcal{A}$ and unselected bins $\mathcal{J} \setminus \mathcal{S}$;
*Output:* A bin index $b$ (may be null if no such bin may be found) with a corresponding packing;
**for** all $j \in \mathcal{J} \setminus \mathcal{S}$ **do**
  **if** $a_{ij} < p_i$ **then**
    Initialize the total cost of bin $j$ to $c_j = f_j + a_{ij}$;
    Update the residual capacity $V_j^* := V_j^* - v_i$;
    Sort the items list $\mathcal{I} \setminus \mathcal{A} \setminus \{i\}$ in non-decreasing order of $(f_j + a_{i'j})/(V_j^* - v_{i'})$;
    **for** all $i' \in \mathcal{I} \setminus \mathcal{A} \setminus \{i\}$ **do**
      **if** $v_{i'} \leq V_j^*$ and $a_{i'j} < p_{i'}$ **then**
        Load $i'$ into $j$;
        $c_j := c_j + a_{i'j}$; $V_j^* := V_j^* - v_{i'}$;
      **end if**
    **end for**
  **end if**
**end for**
Set $b$ to be the bin such that $c_b = \min\{c_j : j \in \mathcal{J} \setminus \mathcal{S}\}$;
Return $b$.

to the total cost (and decreasing order of their capacities in case of equal potential contributions), and then proceed sequentially to fill up each bin as best as possible given the available items. Similarly to HS1, bin-centric heuristics complete the work by, first, topping-up the selected bins with yet unassigned items (sorted in increasing value of their unit impact costs $\bar{a}_{ij}$), second, swapping unassigned items with assigned ones where possible, and third fitting still unassigned items into item-specific spot-market bins. The pseudo-code of HS2 is displayed in Algorithm 3, while Algorithm 6 for HS3 is displayed in the Appendix.

The difference between HS2 and HS3 lies in how the bin potential for contribution is computed. HS2 computes it as the $f_j/V_j$ yielding a linearized unit loading cost for each bin $j \in \mathcal{J}$. HS3, on the other hand, includes the best item-to-bin assignment cost in the computation, i.e., given all the available items for each bin. Both heuristics then fill up each bin by fitting in unassigned items, sorted in increasing value of their unit impact costs $\bar{a}_{ij}$, when capacity allows it and the item-to-bin assignment cost is lower than the cost of assigning to a spot-market bin. Note that this filling-up stage is needed for HS3 as items are no longer all available at all iterations.

**Algorithm 3** HS2

*Input:* Sets of items $\mathcal{I}$ and bins $\mathcal{J}$;
*Output:* Sets $\mathcal{A}$ of items assigned to bins and $\mathcal{S}$ of selected bins;
Initialize $\mathcal{S} = \emptyset$, $\mathcal{A} = \emptyset$;
Sort the bins in $\mathcal{J}$ in increasing order of $f_j/V_j$;
**for** all $j \in$ sorted $\mathcal{J}$ **do**
  Sort the items in $\mathcal{I}$ in increasing order of $\bar{a}_{ij}$;
  **for** all $i \in$ sorted $\mathcal{I}$ **do**
    **if** $v_i \leq V_j$ and $a_{ij} < p_i$ **then**
      Load $i$ into $j$
      $\mathcal{A} := \mathcal{A} \cup \{i\}$, $V_j := V_j - v_i$;
    **end if**
  **end for**
  **if** $j$ is non-empty **then**
    $\mathcal{S} := \mathcal{S} \cup \{j\}$;
  **end if**
**end for**
# Top-up the selected bins if possible;
**for** all bin $j \in \mathcal{S}$ with residual capacity $V_j^*$ **do**
  Sort the items in $\mathcal{I} \setminus \mathcal{A}$ in increasing order of $\bar{a}_{ij}$;
  **for** $i \in$ sorted $\mathcal{I} \setminus \mathcal{A}$ **do**
    **if** $V_j^* - v_i \geq 0$ **then**
      Load $i$ into $j$
      $\mathcal{A} := \mathcal{A} \cup \{i\}$, $V_j^* := V_j^* - v_i$;
    **end if**
  **end for**
**end for**
Look for possible swaps between assigned items and unassigned ones - See Algorithm 1;
Assign any still unassigned item to a spot-market bin.

# 6 Heuristic algorithms for the multi-period 1/V/V/S/N[I2B] model

We propose four constructive heuristic algorithms, *HM1* - *HM4*, for the 1/V/V/S/N[I2B] model. The four heuristics follow the same framework, which is detailed in Section 6.1, the implementation differences between these algorithms being discussed in Section 6.2.

## 6.1 The main framework for the heuristic algorithms (HM)

The proposed framework for the 1/V/V/S/N[I2B] heuristics, displayed in Algorithm 4, consists of the following main steps:

1. Redefine the model by duplicating the bin and item selection variables for each time period of their respective availability intervals;
2. Decompose the problem by time period by relaxing the constraints enforcing that items and bins be selected exactly once and at most once, respectively, within their availability time intervals;
3. Construct a feasible solution by iteratively solving the 1/V/V/S/1[I2B] subproblems independently;
4. Based on the current solutions obtained, identify the period when specific bins can be selected in the most cost-efficient way, assign the items based on this selection, update the sets of unassigned bins and items, and repeat.

Let $\mathcal{I}_t = \{i \in \mathcal{I} : \mathcal{T}_i \cap \{t\} \neq \emptyset\}$ be the set of items available at period $t = 1, \ldots, T$. Similarly, let $\mathcal{J}_t = \{j \in \mathcal{J} : \Gamma_j \cap \{t\} \neq \emptyset\}$ be the set of bins available at period $t = 1, \ldots, T$. The duplication of items and bins at the periods of their availability intervals allows to decompose the problem by time period for the schedule length. Each of the resulting single-period 1/V/V/S/1[I2B] subproblem, corresponding to a period $t = 1, \ldots, T$, may then be addressed using one of the three heuristics presented in Section 5 to assign the items in $\mathcal{I}_t$ to the bins in $\mathcal{J}_t$.

Notice that, at this point, the solution combining those of the independent 1/V/V/S/1[I2B] subproblems is most certainly infeasible for the 1/V/V/S/N[I2B], since some bins may have been selected at more than one time period and some items may also have been assigned to bins at more than one time period. Consequently, the next step determines the *most-efficient* time period for selecting and packing each bin in terms of impact on the overall total cost. Let $\mathcal{A}_t(j)$ be the set of items assigned to (the copy of) bin $j$ at period $t \in \Gamma_j$, and $p_i^* = \min\{p_i^t : \forall t \in \mathcal{T}_i\}$ be the minimum item-to-spot-market-bin cost for items $i \in \mathcal{A}_t(j)$. We may now define $q_j^t = \sum_{\mathcal{A}_t(j)} p_i^*$ as the total cost of shipping the items loaded into bin $j$ at time period $t$ via the spot market. We then define $t_j^*$, the most-efficient time period to select bin $j$, as the period when selecting the bin would have the most significant impact on the overall total cost by reducing the most the spot-market cost one would otherwise have to pay. This procedure is displayed as Algorithm 5.

Once, the most-efficient time period is determined for a bin, together with the corresponding packing, its selection for any other time period is canceled, and the sets of bins and items still to be decided are updates. The process is repeated until there are

**Algorithm 4** Framework for the multi-period (HM) heuristics

*Input:* Sets of items $\mathcal{I}$ and bins $\mathcal{J}$;
Initialize the sets of selected bins for each time period $\mathcal{S}_t = \emptyset,\ t = 1, \ldots, T$;
Initialize the sets of assigned items for each time period $\mathcal{A}_t = \emptyset,\ t = 1, \ldots, T$;
**while** ($\mathcal{I} \neq \emptyset$) and (one item packing may be made) **do**
  **for** t = 1, . . . , T **do**
    Solve the 1/V/V/S/1[I2B] with item set $\mathcal{I}_t$ and bin set $\mathcal{J}_t$;
  **end for**
  **for** all $j \in \mathcal{J}$ **do**
    Identify the time period $t_j^*$ by using *MostEfficient*$(j, t^*)$;
    Update the set of bins selected at time $t^*$, i.e., $\mathcal{S}_{t^*} := \mathcal{S}_{t^*} \cup \{j\}$;
    Update the set of items yet to be assigned at period $t^*$, i.e., $\mathcal{I}_{t^*} := \mathcal{I}_{t^*} \setminus \mathcal{I}_{t^*}(j)$, where $\mathcal{I}_{t^*}(j)$ is the set of items loaded into bin $j$;
    Update the set of items assigned at time $t^*$, i.e., $\mathcal{A}_{t^*} := \mathcal{A}_{t^*} \cup \mathcal{I}_{t^*}(j)$;
    **for** all $t \in \Gamma_j \setminus \{t^*\}$ **do**
      Update the set of items yet to be assigned at period $t$, i.e., $\mathcal{I}_t := \mathcal{I}_t \setminus (\mathcal{A}_{t^*} \cap \mathcal{I}_t(j))$;
    **end for**
  **end for**
  Update the set $\mathcal{I} := \mathcal{I} \setminus \cup_{t=1,\ldots,T} \mathcal{A}_t$;
  Update the set $\mathcal{J} := \mathcal{J} \setminus \cup_{t=1,\ldots,T} \mathcal{S}_t$;
**end while**
Top-up the selected bins as possible - See Algorithm 3;
Look for possible swaps between assigned and unassigned items - See Algorithm 3;
Assign any still unassigned item to a spot-market bin at the period with the smallest spot cost.

no more unassigned items left or it becomes more costly to assign them to a bin than to ship them via the spot market. Similar to the previous heuristics, the last phase of the procedure proceeds to top-up the selected bins with yet unassigned items, then swap unassigned items with assigned ones where possible, and finally fit still unassigned items into item-specific spot-market bins.

## 6.2 The four heuristic algorithms

The heuristic framework proposed above for the 1/V/V/S/N[I2B] may be implemented differently according to 1) the order of examining the bins to find their most-efficient time period, and 2) the heuristic used to address the single-period subproblems. Two cases are defined for the former case: no ranking, and ranking the bins by cost contribution. The HS1 - HS3 heuristics (Section 5) may be called upon to address the 1/V/V/S/1[I2B] subproblems.

**Algorithm 5** *MostEfficient*$(j, t_j^*)$ procedure to determine the best time period to select bin $j$

**for** all $t \in \Gamma_j$ **do**
  Delete any item in $j$ already assigned to another bin;
  Top-up the non-empty bin $j$ with unassigned items;
  Compute $q_j^t$;
**end for**
Set $t_j^*$ to the time period such that $q_j^{t^*} = \max\left\{q_j^t : t \in \Gamma_j\right\}$;
Return $t_j^*$.

We define four heuristics for the multi-period problem, by combining these possibilities:

1. ***HM1***: HS1 heuristic (Algorithm 1) for the single-period subproblems; Bins are not ranked;
2. ***HM2***: HS2 heuristic (Algorithm 3) for single-period subproblems; Bins are not ranked;
3. ***HM3***: HS3 heuristic (Algorithm 6) for single-period subproblems; Bins are not ranked;
4. ***HM4***: HS3 heuristic (Algorithm 6) for single-period subproblems; Bins are ranked: the $q_j^t$ values are computed for all bins $j$ and time periods $t$, and bin $j^*$ is selected at time period $t^*$ for which $q_{j^*}^{t^*} = \max\left\{q_j^{t^*} = \max\left\{q_j^t : t \in \Gamma_j\right\} : j \in \mathcal{J}\right\}$.

# 7 Computational Results

We present and analyze the outcome of the computational experiments that we carried out for the proposed mathematical models and heuristic algorithms. Four goals were fixed to these experiments. First, we aimed to evaluate the performance of the proposed heuristics by measuring 1) the quality of the corresponding upper bounds in terms of the optimality gap, and 2) the computational efficiency in terms of CPU time in seconds. To compute the optimality gaps and evaluate the computational performance, the instances were solved using the MIP solver of *CPLEX version 12.6.3* IBM (2016). Section 7.2 is dedicated to this analysis. The second goal, Section 7.3, was to analyze the impact of a number of problem attributes on the behavior of the solution approach we propose, the tightness of the availability time windows, in particular.

The third objective was to assess the ability of the heuristics to improve the performance of commercial solvers by feeding the heuristically-obtained feasible solutions to

an exact MIP solver. More precisely, we set the objective function value obtained by the heuristic algorithm as the initial cutoff upper bound within the commercial solver, so that, hopefully, more nodes (with objective values at or above the upper bound) are pruned faster. We initiated this experiment motivated by our own results as well as by the literature reporting on the inefficiency of commercial solvers in solving bin packing models Crainic et al. (2011); Baldi et al. (2012); Correia et al. (2008). Notice that this is not meant to diminish the capabilities of commercial solvers for most MIP problem settings. It is known, in fact, that such solvers, which are not designed for specific problem types, can be used to develop tailored solution methods to specific problem settings. Our results, discussed in Section 7.5, support this claim by showing that providing a good heuristic upper bound value greatly improves the computational performance of CPLEX.

Finally, we evaluate in Section 7.4 the interest of accounting explicitly for the time dimension of the problem and of the methodology we propose to address this issue. We perform this analysis by measuring the reduction in total cost obtained by solving the multi-period model, with respect to a myopic approach which solves $T$ single period problems, each considering the items and bins first appearing at that period.

These experiments were conducted for the most part on four sets of randomly generated data, two sets for the single-period 1/V/V/S/1[I2B] heuristics, and two sets for the multi-period 1/V/V/S/N[I2B] ones. A total of 1300 test instances were thus generated, aiming to represent various situations with respect to the item sizes relative to the bin sizes. Section 7.1 describes the test instances and their generation process.

All the algorithms were coded in the C programming language and were compiled with *gcc version 4.8.1* GNU (2018). The computational platform used to run the heuristic algorithms was a machine with 8GB of memory and a processor at 1.8 GHz, while the computational platform used for solving the instances to optimality was the supercomputer "CEDAR" managed by "Calcul Québec" and "Compute Canada". We used a single node on this cluster with a 2.26 GHz processor. The memory requested to solve the problems was 8 GB for all the instances with a time limit of 1 hour. Furthermore, we set the internal parameter of parallel CPLEX (`CPX_PARAM_THREADS`) to 1 to use one thread only. We left the cut generation parameters to their default values so that CPLEX can generate as many cutting planes as needed. Note that throughout this section, we assume that CPLEX has found an optimal solution for an instance when the optimality gap is 0.05% or less.

## 7.1 Description of the test instances

As indicated above, four sets of instances were generated, two each for the cases considered, i.e., the 1/V/V/S/N[I2B] and the 1/V/V/S/1[I2B] problems, respectively. The same idea guided the generation for both cases, except that 1/V/V/S/N[I2B] instances

were generated over multiple periods. Therefore, the instances of the first set, Set 1 in the following, represent in both cases situations where the item sizes are relatively small compared to the bin capacities. Similarly, instances of Set 2 represent situations where there is a mix of large and medium size items compared to the capacities of the bins. Note that these instances are similar in spirit with the instances used by Baldi et al. (2012); Crainic et al. (2011), but bin capacities are randomly generated. Note also that the C-code employed to generate the instances used for the experiments in this paper can be found in Fomeni et al. (2020).

The 1/V/V/S/N[I2B] instances were generated using the following parameters:

- Availability time window for items and bins: Randomly chosen initial period between 0 and $T$, with length randomly varying between 3 and 6 time periods;
- Fixed cost for bin $j$: $f_j = 100\sqrt{V_j}$;
- Item-to-bin assignment cost $= a_{ij}^t = \beta_1 + \beta_2 \times (t - \underline{t}_i) + \beta_3 \times \max\{(\gamma_i - t), 0\} + \beta_4 \times \max\{(t - \delta_i), 0\}$, for item $i \in \mathcal{I}$, bin $j \in \mathcal{J}$, and time period $t \in \mathcal{T}_{ij}$, where $\beta_k, k = 1, 2, 3, 4$ are random numbers in the intervals $[20, 30], [3, 5], [1, 2]$ and $[3, 5]$, respectively;
- Spot market cost $= p_i^t = p_1 + \beta_2 \times (t - \underline{t}_i)$, where $p_1$ is a random number between $[100, 200]$ and $\beta_2$ is the same as in the assignment cost.

The values for the other parameters are given in Table 1. Combinations of these values define five and six types of instances for Sets 1 and 2, respectively, as displayed in the table. Ten instances were randomly generated for each type and number of items, for each set.

The cost parameters defined above were also used for the 1/V/V/S/1[I2B] instances. The instance types and the corresponding parameters are displayed in Table 2.

## 7.2 Performance of the heuristics

The objective of this section is to analyze the efficiency of the heuristic algorithms proposed for the single, Section 7.2.1, and multi-period, Section 7.2.2, cases, respectively. Recall that these algorithms are constructive heuristics, which aim to produce a good feasible solution in a very short amount of time. They may thus be directly applied to address practical problems, or they can yield cutoff upper bounds to enhance the performance of commercial solvers (Section 7.5).

Table 1: Instance types, multi period: 1/V/V/S/N[I2B]

| Set 1 Types | N. Items = 100,200,300,500,750,1000 | | | |
|---|---|---|---|---|
| | N. Bins | N. Periods | Item Vol. | Bin Cap. |
| T1 | [10,100] | 10 | [1,20] | [50,250] |
| T2 | [20,100] | 20 | [1,20] | [50,250] |
| T3 | [50,100] | 30 | [1,20] | [50,250] |
| T4 | [50,150] | 10 | [1,20] | [50,250] |
| T5 | [50,200] | 15 | [1,20] | [50,250] |
| Set 2 Types | N. Items = 100,200,300,500,750,1000 | | | |
| T1 | [75,100] | 10 | [1,100] | [20,250] |
| T2 | [100,150] | 10 | [20,100] | [20,250] |
| T3 | [150,200] | 10 | [50,100] | [20,250] |
| T4 | [75,100] | 20 | [1,100] | [20,250] |
| T5 | [100,150] | 20 | [20,100] | [20,250] |
| T6 | [150,200] | 20 | [50,100] | [20,250] |

Table 2: Instance types, single period: 1/V/V/S/1[I2B]

| Set 1 Types | N. Items = 100,200,300,400,500,600,700,800,900,1000 | | | |
|---|---|---|---|---|
| | N. Bins | N. Periods | Item Vol. | Bin Cap. |
| T1 | [10,100] | 10 | [1,20] | [50,250] |
| T2 | [10,100] | 20 | [1,20] | [200,250] |
| Set 2 Types | N. Items = 100,200,300,500,750,1000 | | | |
| T1 | [75,100] | 10 | [1,100] | [50,250] |
| T2 | [100,150] | 10 | [20,100] | [50,250] |
| T3 | [150,200] | 10 | [50,100] | [50,250] |
| T4 | [50,100] | 20 | [1,100] | [100,200] |
| T5 | [100,150] | 20 | [20,100] | [100,200] |
| T6 | [150,200] | 20 | [50,100] | [150] |

The tables of this section follow the same format in displaying results by instance type (Column 1) and cardinality of the item set (Column 2). Namely, the average, over the 10 instances, of the relative optimality gap with respect to the optimal solution (OPT) of CPLEX and the average CPU time (in seconds) are displayed for each heuristic included in the respective table. Notice that, CPLEX could not find an optimal solution in all cases within the 3600 seconds time limit, in particular for the multi-period problem setting. In those cases, the best lower bound (LB) obtained by CPLEX was used to compute the relative gap (Equation 3).

$$relative\ heuristic\ gap = \left( \frac{Heuritic\ Solution - CPLEX\ LB \text{ or } OPT}{CPLEX\ LB \text{ or } OPT} \right) \times 100. \quad (3)$$

The optimality gaps and the CPU computing times are displayed for the branch-and-cut algorithm of CPLEX in the BnC set of columns. When optimality was not reached within the time limit, the results in Tables 5 and 6 for the multi-period case display additional information: the number of instances (out of ten) that were solved by CPLEX (Column #Sol.) and the number of instances for which at least one feasible solution was found (Column #Feas.) within the time limit. Note that the average optimality gap of CPLEX in those cases was computed for the instances for which a feasible solution was found, only.

We further support the conclusions of the following analyzes by plotting, in boxplot form, the spread of the average gaps and CPU times achieved by the proposed heuristics and by CPLEX for the single and the multi-period formulations in Figures 1 and 2, respectively.

### 7.2.1 Single-period problem

The results obtained for assessing the quality of the solutions provided by the three heuristics for the **1/V/V/S/1[I2B]**, HS1, HS2, and HS3, are shown in Table 3 for the instances in Set 1 and in Table 4 for the instances in Set 2. All gaps were computed relative to the CPLEX lower bound as shown in Equation (3). Figure 1 displays the overall spread of the average gaps and CPU times for these instances.

In terms of computational times, one notes that all three heuristics, HS1, HS2, and HS3, require a few seconds only to solve most instances. The average computational times of HS1 for instances with more than 500 items are 5 seconds or less, while it requires less than half a second for instances with less than 500 items.

One notes that, while CPLEX is able to find feasible solutions within 0.15% of optimality, the computational times required are rather large. For example, instances in the Type 2 group require up to 500 seconds, while most of those in the Type 1 group

Table 3: Performance for on Set 1 instances

| | | HS1 | | HS2 | | HS3 | | BnC | |
|---|---|---|---|---|---|---|---|---|---|
| | $\|\mathcal{I}\|$ | Gap | Time | Gap | Time | Gap | Time | Gap | Time |
| Type 1 | 100 | 10.99 | 0.01 | 7.56 | 0.01 | 7.69 | 0.01 | 0.00 | 32.80 |
| | 200 | 9.75 | 0.06 | 7.38 | 0.06 | 7.94 | 0.11 | 0.01 | 260.43 |
| | 300 | 8.41 | 0.17 | 9.09 | 0.18 | 13.63 | 0.33 | 0.01 | 1,328.06 |
| | 400 | 6.91 | 0.44 | 10.40 | 0.40 | 12.67 | 0.76 | 0.04 | 2,970.02 |
| | 500 | 6.66 | 0.62 | 17.31 | 0.57 | 19.41 | 1.05 | 0.05 | 2,345.45 |
| | 600 | 6.65 | 0.90 | 18.88 | 0.79 | 18.02 | 1.49 | 0.05 | 2,608.62 |
| | 700 | 6.49 | 1.72 | 16.89 | 1.52 | 16.96 | 2.96 | 0.10 | 2,732.17 |
| | 800 | 6.43 | 3.38 | 13.09 | 3.10 | 15.20 | 6.09 | 0.12 | 3,600.07 |
| | 900 | 6.56 | 3.04 | 16.91 | 2.91 | 16.55 | 5.65 | 0.11 | 3,240.27 |
| | 1000 | 6.83 | 5.79 | 16.22 | 5.35 | 16.10 | 10.23 | 0.15 | 3,600.07 |
| Type 2 | 100 | 6.25 | 0.01 | 5.87 | 0.01 | 5.87 | 0.01 | 0.00 | 1.58 |
| | 200 | 3.44 | 0.06 | 4.40 | 0.06 | 4.40 | 0.11 | 0.00 | 11.59 |
| | 300 | 3.15 | 0.17 | 5.67 | 0.16 | 5.67 | 0.32 | 0.01 | 13.12 |
| | 400 | 3.60 | 0.39 | 3.52 | 0.38 | 3.52 | 0.73 | 0.01 | 33.94 |
| | 500 | 3.12 | 0.54 | 18.25 | 0.53 | 18.25 | 1.02 | 0.02 | 535.34 |
| | 600 | 4.00 | 0.77 | 20.82 | 0.75 | 20.82 | 1.45 | 0.00 | 92.08 |
| | 700 | 3.20 | 1.55 | 16.14 | 1.50 | 16.14 | 2.90 | 0.00 | 122.21 |
| | 800 | 3.79 | 3.13 | 12.39 | 3.03 | 12.39 | 5.83 | 0.00 | 311.00 |
| | 900 | 4.34 | 2.91 | 19.66 | 2.82 | 19.66 | 5.45 | 0.00 | 502.09 |
| | 1000 | 4.24 | 5.59 | 16.71 | 5.23 | 16.71 | 10.08 | 0.00 | 586.34 |

require up to the 3600 seconds limit to get to this optimality gap. This may be much too long in many applications, considering that bin packing problems are often solved at the operational level where the time available to make decisions is often quite limited Crainic et al. (2011). This is achievable by one of our proposed heuristics, HS1, which is able to produce good quality solutions within less than five seconds. Moreover, given that the computational times of HS1 are so small, it means that this heuristic algorithm is a good candidate for generating starting solutions that may be used by commercial solvers. We show in Section 7.5 how these solutions are able to dramatically improve the computational time of CPLEX for the difficult instances in Set 1. The illustration of the average CPU times in Figure 1a (right-hand side image) supports this analysis.

Turning now to solution-quality performance, the results for the instances of Set 1, reported in Table 3, indicate that HS2 and HS3 behave similarly but are not competitive for these instances, with average optimality gaps near 20% in some cases. HS1 has a very good performance for this set of instances, however, as it consistently finds solutions within 6% of optimality for most instances of Type 1 and within 3% of optimality for most instances of Type 2. On the other hand, one notes that the Branch-and-Cut algorithm of CPLEX is unable to optimally solve all the instances within the 3600 seconds time limit, but it identifies feasible solutions within 0.15% of optimality for all the instances. Most of the highest gaps are observed for the Type 1 instances, which are also the instances for

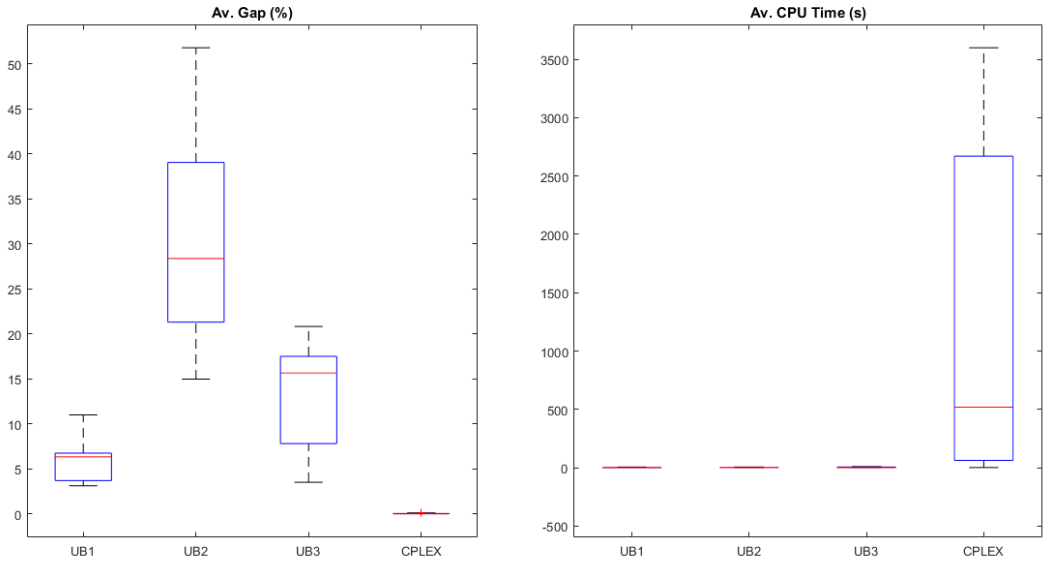
which the optimality gap of our best proposed heuristic, HS1, is above 6% of optimality.

The last observation suggests that the instances in this group have characteristics, symmetry in particular, which make them harder to solve for both CPLEX and our algorithms. Indeed, as already observed in previous studies, symmetry is induced in 1/V/V/S/1 problems by the presence of many items with small volumes relative to the capacity of the available bins. Then, replacing an item in a bin with another one of similar size makes generally no difference in the loading pattern and the value of the objective function. Exact and heuristic algorithm then need significantly broader explorations before identifying good solutions, with notable performance degradation. A good heuristic, which may also accelerate exact solution methods, is then an interesting choice. The proposed HS1 heuristic addresses this challenge, as illustrated in Figure 1a.

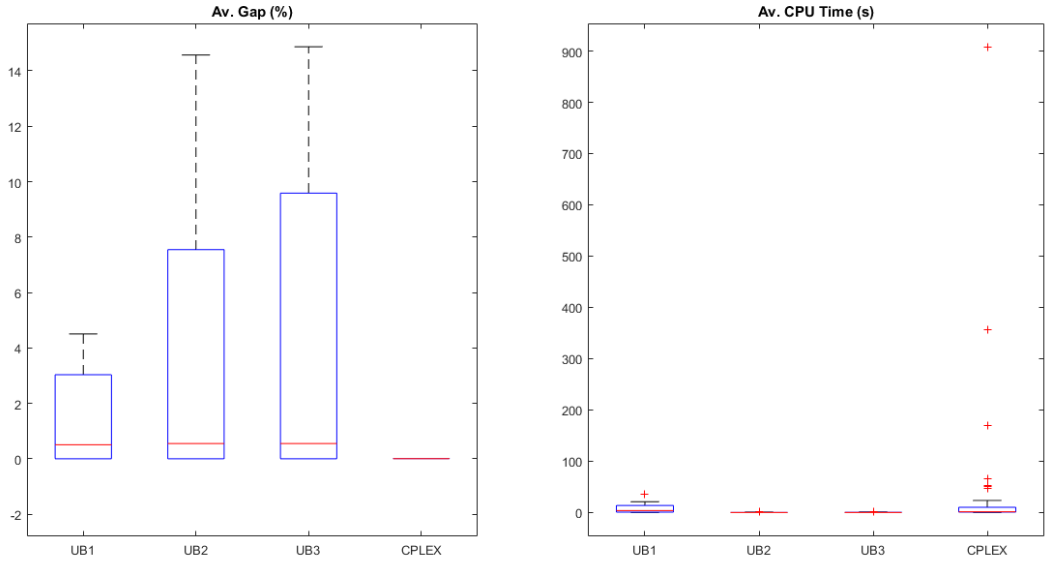
To further analyze the behavior of the heuristics, one observes in Table 3 that the optimality gaps obtained by the HS2 and HS3 heuristics tend to increase with the number of items. The HS1 heuristic offers a superior performance. Indeed, while the optimality gaps are higher for instances with 100 and 200 items, they not only significantly outperform those of the other two heuristics, but they also become almost stable for larger instances. This behavior may be related to the structure of the problem and that of the algorithms. On the one hand, HS2 and HS3 are bin-centric as they focus on selecting a bin first, then looking for the combination of items to be loaded into the bin. Thus, it is relative easy to find a good combination for each selected bin when there are few items, while with a larger number of items, the problem becomes more complex. On the other hand, HS1 is item-centric as it focuses on finding the best bin for each item. This, combined with the fact that instances in Set 1 are characterized by items with small sizes to be loaded into larger bins, may result, when there are only few items, in some bins being opened up with few opportunity of being filled up. When a large number of items is available, however, each bin that is opened is more likely to be filled-up, thus yielding solutions that are much closer to optimality.

Item dimensions are more varied with respect to bin volumes in Set 2 instances compared to those of Set 1, many instances including large items. This reduces the symmetry resulting in better-performing solution methods, as illustrated by the figures in Table 4 and Figure 1b. One thus notices that CPLEX efficiently solves the instances of Set 2. It consistently finds the optimal solutions, computational times exceeding 50 seconds only for a handful of instances.

Very good performance may be also observed for all heuristics on the Set 2 instances, computational times for the three heuristic algorithms never exceeding 36 seconds. HS1 appears again as the best performing of the three heuristics, with optimality gaps varying between 0.00% and 4%. Notice that optimality gaps are low in most cases, except for Type 1 and Type 4 instances with low item volumes compared to the bin volumes. This reinforces the discussion above on instances more difficult to solve when including items




(a) Set 1




(b) Set 2

Figure 1: Spread of average Gaps and CPU times for

of small size. Figure 1b illustrates these findings.

In summary, the results of the computational experiments reported in this section show that the proposed model may sometimes be solved to optimality using commercial solvers. This is mainly the case for instances with large-size items with respect to bin capacity, or a mix of large and medium size items. In the case of small-size items, the CPLEX solver finds it difficult to solve the instances. However, one of our proposed heuristics, HS1, proves to be able to provide a great trade-off between computational time and optimality gap regardless of the type of problem at hand. HS1 displays useful traits for applications to practical problems. First, it has the ability to produce good quality solutions in a very small amount of time, which is extremely important in practical applications. Second, it is also a good candidate for generating starting solutions to speed-up the performance of commercial solvers.

### 7.2.2 Multi-period problem

We turn to analyzing the performance of the heuristic algorithms HM1, HM2, HM3, and HM4 (Section 6.2) proposed to address the multi-period 1/V/V/S/N[I2B]. These experimental results are presented in Tables 5 and 6 for instances of Sets 1 and 2, respectively, and illustrated in Figure 2.

Recall that instances in Set 1 have items of rather small size compared to the size of the available bins which, as observed previously, increases the symmetry of these problems and makes them somewhat difficult to address. Notice, moreover, that the multi-period nature of the problem tends to increase this symmetry and the time required to address it, by increasing the assignment possibilities for items to bins in several periods. Such combinations of characteristics, which is further analyzed in Section 7.3, challenges all methods, in particular ways, illustrated in the results reported in Table 5 and Figure 2a.

With respect to solution quality, one observes that the HM4 heuristic outperforms the other three, which display a somewhat similar behavior. As for computational efficiency, all heuristics are extremely fast. Thus, e.g., HM4 consistently provides solutions within 6% of optimality in less than 15 seconds. This behavior contrasts with that of CPLEX, which struggles for these instances in terms of both optimality gap and computational time. The displays of Figure 2a, as well as the last two columns of Table 5, are particularly telling in this respect.

One notices, first, that CPLEX struggles to find the optimal solutions in many cases (some 194 instances), while not even finding a feasible one for 41 instances. Given the fact that the average optimality gaps are low, the second conclusion is that it is the high symmetry of this class of instances which makes pruning the branch-and-bound tree tedious and induces this behavior. Comparing the heuristics and the solver, one also

notices that not only all heuristics find feasible solutions to all instances, but also that they do so in very short computational times. Moreover, using CPLEX as a heuristic by stopping the search after 10 minutes (Column $Gap^{10}$) does not appear enticing when compared to HM4. As discussed in Section 7.5, using HM4 to provide a good initial upper bound increases the performance of the exact solver.

The results for the instances of Set 2, reported in Table 6 and Figure 2b, indicate that the proposed heuristics perform well, in terms of solution quality and computational efficiency, with heuristic HM1 dominating. Indeed, on the difficult (low item volumes with respect to bin sizes) Type 1 and Type 4 instances, HM1 yields gaps between 0.6% and 3%, while the gap does not exceed 0.18% on all other instance types. And these solutions are obtained very fast, significantly faster than CPLEX. It is noteworthy that, even on these instances which appear easier to address, CPLEX still fails on a number of cases.

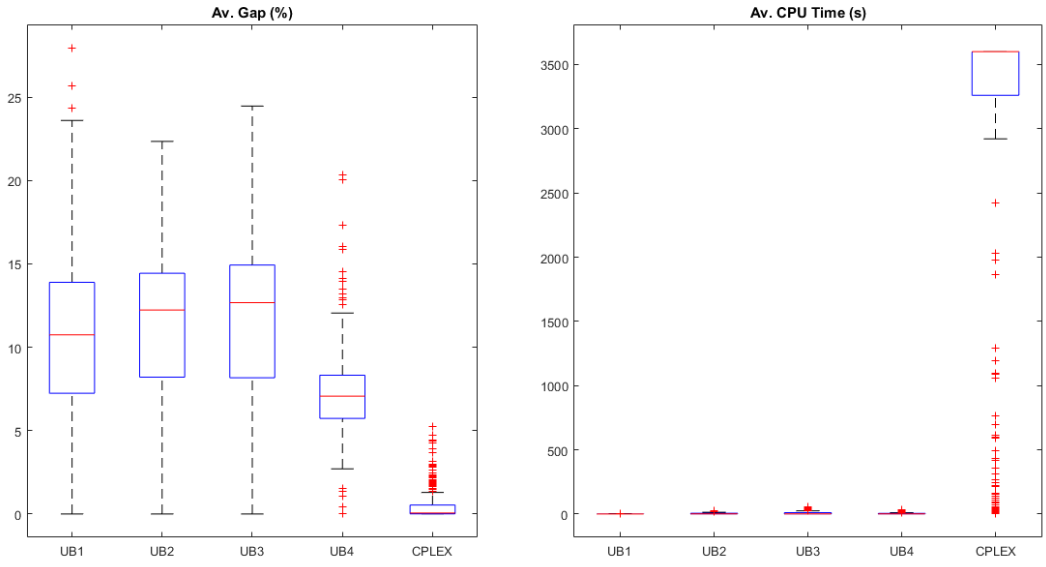
In summary, the computational experiments reported in this section also highlight the practical usefulness of the proposed heuristic algorithms. More precisely, it has been observed that, even thought CPLEX can be quite efficient on certain instances, it struggles when problem symmetry is high and when the size of the instances increases. It actually starts failing to prove optimality and even to find feasible solutions. The corresponding increases in computational times are not nearly as pronounced for the heuristics, completing the search under a minute in most cases. Moreover, heuristics are quite robust, producing high-quality solutions on all instances.

Two final remarks before proceeding to analyze the impact of a number of parameters on the multi-period solutions. First, a different heuristic stands up among those proposed for the different types of problem instance studied. These may be used as stand-alone solution methods or, as discussed in Section 7.5, to speed-up the performance of commercial solvers. Second, we attained our goal of efficient heuristics for the the 1/V/V/S/N[I2B]. Moreover, given the fact that all proposed heuristics provide good solutions in short computational times, one could enhance performance in application contexts by applying all of them and selecting the most appropriate solution.

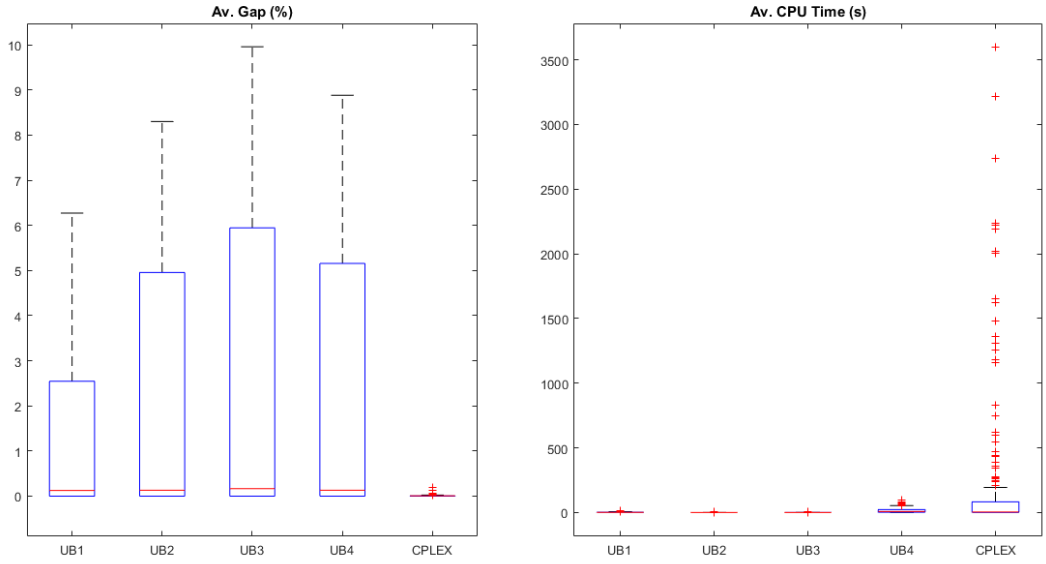
## 7.3 Parameter impact on multi-period solutions

This section is dedicated to a further exploration of the impact of a number of problem attributes, e.g., number of items, bins, and periods, and tightness of the availability time windows, on the performance of the heuristics proposed for the 1/V/V/S/N[I2B] and the solutions they provide.

We start the discussion with two observations that, we believe, highlight the fact that the time dimension of the 1/V/V/S/N[I2B] is highly important and cannot be dissociated neither from the model, nor from the algorithm design.




(a) Set 1




(b) Set 2

Figure 2: Spread of average Gaps and CPU times for 1/V/V/S/N[I2B]

First, one notices that the performance of the four heuristics for the 1/V/V/S/N[I2B] on instances with low item volumes compared to the bin volumes (Set 1 and Types 1 and 4 of Set 2) tends to improve greatly (significant decrease in optimality gaps) as the number of items increases. This contrasts with the observation made for the similar instance types for the single-period , and could even seem to be counter-intuitive. This behavior can be explained, however, by the fact that the items and the bins in a multi-period problem setting have availability time intervals that may overlap, thus offering more opportunity for consolidation. And this opportunity increases with the number of items and compensates for the inherent symmetry of the problem. We further analyze the impact of the availability time windows of bins and items later in this section.

The second observation is that a different heuristic dominates for the multi-period (HM4) and single-period (HS1) problem settings. Given the similarities between the instances in most attributes except the time factor, we believe that the fact that the HM1 heuristic does not behave as well as HS1 on which it is based, underscores the importance of addressing time in models and algorithms alike.

The number of items appears less critical when instances include a good number of large items or a mix of medium and large-sized items, always with respect to the bin volumes. This is not surprising as the loading combinatorics decreases with larger item sizes (fewer larger items may be packed in the bins of the "same" size and, thus, fewer loading combinations are possible).

Turning now to the impact of the number of periods and bins, we first observe that, as the instances were generated based on different ranges of bin volumes, trends are better observed by comparing the average values obtained. We then first examine the instance types T1, T2, and T3 of Set 1. The range of the number of bins for these instance types decreases with the number of periods, due to an increasingly higher minimum number of bins, resulting into increasing numbers of bins per instance. Intuitively, this should create more opportunities for consolidated packing (more periods to do it into more bins available). The intuition is confirmed by the results displayed previously, as optimality gaps decrease drastically, for all numbers of items. Second, we compare the behavior for instance T1 versus T4 of Set 1. We observe that the optimality gap tends to increase with the number of bins, but a closer analysis indicates that this happens only when the number of items increases as well. As this is observed for all heuristics and CPLEX, one then concludes that the impact is not due to the number of bins but rather the increasing symmetry due to the number of low-size items. We finally turn to the instances of Set 2, comparing pairs of instance types according to the range of the number of bins, i.e., T1-T4, T2-T5, T3-T6. No significant difference is observed, except a slight improvement with the number of bins noticed for all numbers of items. It seems that the proposed heuristics are robust with respect to the number of items - number of bins ratio, when instances include large items or a mix of large and medium-sized ones.

We now turn toward the impact of tightening the availability time windows of both bins and items. We focused the experimentation on the difficult 1/V/V/S/N[I2B] instances with low-volume items compared to the bin volumes. The T1 (Set 2) instances are described in Section 7.1, and the corresponding results are found in the first six rows of Table 6. Recall the time-window width for the latter instances was [3,6].

We generated 240 new T1 instances by modifying the width of the item and bin availability time windows as indicated in the first column of Table 7, which reports the results of the experimentation. The table displays for each instance type, number of items, and number of available bin types (over the 10 periods) the average over 10 instances of the CPU times (in seconds) for HM4, the best heuristic identified previously, and CPLEX, as well as the average optimality gaps between the respective CPLEX lower bounds and the HM4 solutions.

The results are interesting. Notice that tightening the availability time windows has a double, potentially conflicting, impact. On the one side, it reduces the overlap periods when "many" items and bins are simultaneously available. This reduces the opportunity for item-to-bin assignment, decreasing the symmetry of the problem, and increasing the computational efficiency. The significant and continuous reduction in CPU times for both solution methods supports this claim. It would appear that reducing the bin time windows has more impact on CPLEX CPU times than reducing the item ones; the reverse seems to be true for the heuristic. The impact on solution quality, on the other hand, is somewhat less obvious. Thus, reducing time windows slightly, from [3,6] to [0,2], appears unprofitable for large numbers of items. Indeed, reducing slightly the item-to-bin assignment opportunities leaves still a sufficiently large search space for a constructive heuristics to end up with local optimum using too many spot bins with high costs. This observation is sustained by the fact that, even though performance improves for the second instance sub-type where the search space is reduced by reducing the bin offer, the heuristic behaves much better when the space is reduced by reducing the item assignment possibilities while still proposing a larger bin offer (third type). When very few assignment possibilities exist (fourth type) the search space is relatively small and the HM4 heuristic identifies very good, almost optimal, solutions.

The current conclusion is thus that, first, availability time windows play a role in the efficiency of the heuristics and exact solution methods, tighter windows improving performances. Second, these experiments showed the excellent behavior of the proposed heuristic, HM4, in all cases studied. Finally, it is noteworthy that, in actual applications, one searches first and foremost for low-cost solutions, and the proposed heuristics are sufficiently fast to identify them. Consequently, one would not recommend to negotiate or fix tighter time windows as the resulting solutions may require relying on more spot bins and, thus, be more costly.

## 7.4 Comparisons with a myopic approach

Recall that the main idea of the problem setting we define and explore, is that it is more appropriate and beneficial to explicitly consider the time characteristics of the problem within a comprehensive model, rather than myopically considering each period individually. More precisely, it is better to solve the multi-period 1/V/V/S/N[I2B] model we propose in this paper, rather than a sequence of single-period models, each with the currently available items and bins. Intuitively, the multi-period approach allows to take advantage of the fact that decisions of some items or bins could be delayed for a better consolidation opportunity in a future time period. Obviously, one should avoid as much as possible that such delays induce penalty costs or missing the delivery due dates. According to our best knowledge, this intuition has not been validated in the literature and, thus, we undertook the experimentation reported in this section.

We first solved each instance in Sets 1 and 2 of 1/V/V/S/N[I2B] with each of the four heuristics proposed, HM1, HM2, HM3, and HM4. We use Cost(MultiP) to identify the objective function-value of any one of those runs. The formulation is then solved for each of the $T$ time periods of the corresponding instance, with the bins and items made available for that time period. The best heuristic for the single-period case, HS1, was used and the objective-function values of the $T$ problems were summed up. Let Cost(Myopic) identify the resulting value of solving $T$ independent problems. We finally computed the estimated improvement as

$$Improvement = \left( \frac{\text{Cost(Myopic) - Cost(MultiP))}}{\text{Cost(Myopic)}} \right) \times 100,$$

We display the results of these comparisons in Table 8, as averages over ten instances, for each instance in Sets 1 (there is no T6 in Set 1) and 2 and each heuristic proposed for the 1/V/V/S/N[I2B].

The obvious observation is that there is no negative value in Table 8. This means that, for all the 660 test instances of Sets 1 and 2, and for all the heuristics proposed, solving the multi-period 1/V/V/S/N[I2B] model is always better than solving a sequence of independent single-period problems. This shows that introducing the possibility to delay the selection of bins and the assignment of items to bins, within the limits of time and the penalty values imposed by the particular problem setting, is beneficial. And the benefits can be very high. These results emphasize that the proposed 1/V/V/S/N[I2B] model is important not only from a theoretical point of view, but that it has the potential of significantly reducing the operational costs for practical applications.

## 7.5 Heuristic solutions to enhance a MIP-solver performance

Given the excellent performance of the proposed heuristics, we wanted to explore the potential benefit of using them to enhance the performance of a commercial solver. More precisely, we aim to show that the solutions provided by the heuristic algorithms may be provided to the MIP solver of CPLEX to effectively reduce computational times. We tested two approaches. First, we used the heuristic solution to warm start the solver and, thus, initiate the exploration. Second, we used the solution value provided by the heuristic as a good initial upper bound for the MIP to enhance pruning and limit the number of branch-and-bound nodes explored.

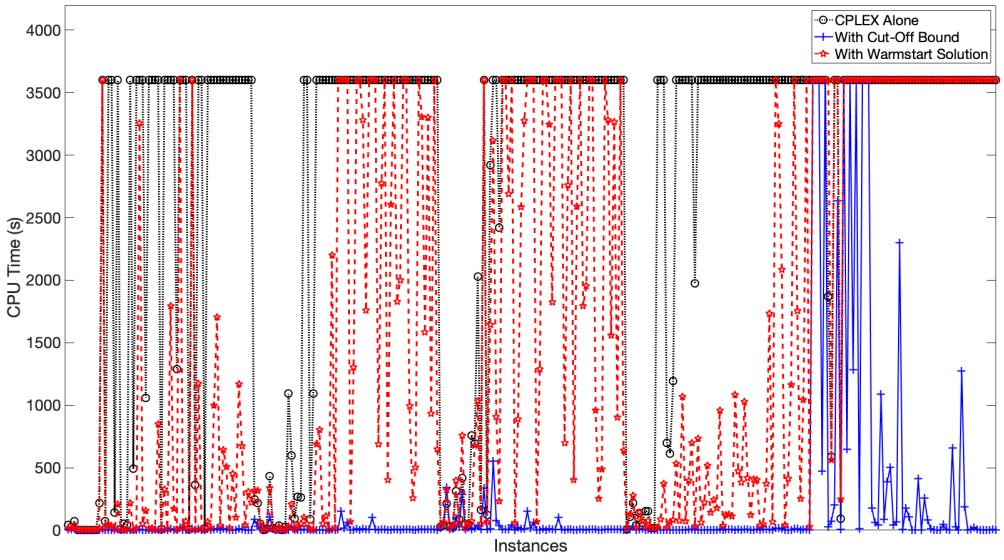



Figure 3: Impact of using the heuristic solution in CPLEX - CPU time comparisons, Set 1 of 1/V/V/S/N[I2B]

We focused this experiment on the difficult instances of the Set 1 for the 1/V/V/S/N[I2B] problem. Each instance was first solved using HM4, the best heuristic for the problem setting. The instance was then solved twice, first by imputing the HM4 solution as warm start to CPLEX and, second, by imputing the objective-function value of the HM4 solution as the initial cut-off upper bound in CPLEX (using the parameter `CPX_PARAM_CUTUP`), meaning that the solver will prune any node that has a lower bound larger than or equal to this value. All the other CPLEX parameter settings were left at the same values as for the previous experiments. The outcome of this experiment is shown in Figure 3, which plots the comparisons between the CPU times of CPLEX alone, with the cut-off upper bound, and with the warm start.

The results illustrate the interest of using the heuristic to initiate the commercial-

solver exploration. For some 60% of the instances tested, using the heuristic solution to warm start the solver improves its computational performance. The solver still fails, however, to solve a good number of instances within the one hour time limit. Moreover, the computational gains observed are less marked than when the heuristic solution value is used to initiate the solver upper bound only. This behavior could be explained by the amount of work the solver dedicates to improving, through various heuristics, externally-obtained feasible solutions.

One can clearly observe from the results displayed in Figure 3, that using the heuristic solution objective-function values as initial cut-off upper bounds enables the commercial solver to reduce its computational times by a significant amount compared to CPLEX alone and to warm-started CPLEX. It's noteworthy to recall that, without the heuristic cut-off capability (Table 5) , CPLEX was unable to find even a feasible solution for 41 instances (out of 300) within the 1 hour time limit. In comparison, using the heuristic solution values as initial upper bounds enables the solver to find a feasible solution for all instances, the 1 hour time limit having been reached for 9 instances only. Moreover, 290 instances out of 300 were solved to optimality within 20 minutes, Similar performance was observed for the 1/V/V/S/1[I2B] problem setting, CPLEX solving all the instances of Set 1 within 40 seconds, when the HS1 heuristic results were input as initial cut-off upper bound.

The results of these experiments further emphasize the usefulness of the proposed heuristic algorithms. illustrating quite clearly how our heuristics can be combined with other solution strategies to enhance computational performance.

# 8 Conclusions

We proposed in this paper a novel time-dependent Bin Packing problem setting, which addresses several shortcomings in the literature and extends the capabilities of the Bin Packing methodology to many practical applications, in particular to logistics capacity planning. This new *Multi-period Variable Cost and Size Bin Packing Problem with Assignment Costs* explicitly considers item and bin-specific item-to-bin assignment costs, which depend on the physical, e.g., item and bin size, economic, e.g., bin selection fixed cost and the cost of item "transported" by the bin, and, especially, the temporal attributes of items and bins. Several temporal attributes were defined, including the availability of bins for selection and utilization and of items for assignment to such a regular bin, the length of time the bin can be used (the so-called travel times), and the length of time items must be kept in a bin (the so-called earliest, latest, and due delivery dates) with associated costs. Moreover, item-specific, in terms of size, spot-market bins could be used at higher cost for the items one cannot fit into the selected bins.

We proposed mathematical formulations for the *Multi-period Variable Cost and Size Bin Packing Problem with Assignment Costs*, as well as for the *Single-period Variable Cost and Size Bin Packing Problem with Assignment Costs*. The later, which may be called static without a look-ahead capability, extends the current Bin Packing literature, and may be used both in certain planning contexts and in heuristics for the multi-period version. The main contribution is brought by the *Multi-period Variable Cost and Size Bin Packing Problem with Assignment Costs* model, addressing time-dependent multi-period problem settings, where decisions are taken "now" for a multi-period schedule length, given data which may be totally (firm contracts) or partially (certain for the current period and predicted for the following ones) known. The multi-period formulation and solution methods may thus be directly used in many practical real-life cases, as well as to address the problem at the current time period in a rolling-horizon approach.

We introduced a set of constructive heuristic algorithms to address these formulations; three for the single-period and four for the multi-period problem settings. The heuristics were evaluated and compared through an extensive computational experimentation involving 1540 problem instances of the single and multi-period problems. The results of these experiments showed the high level of performance provided by the proposed heuristics in terms of CPU computation time and solution quality. Indeed, the proposed heuristics rapidly provide high-quality solutions. Moreover, while a well-known commercial solver was quite efficient on certain instances, its computing times increased rapidly with the problem size. The software failed to even find a feasible solution on a large number of instances. On the other hand, the increase in computing times of the heuristics was rather negligible, the heuristics proving to be quite robust in producing high-quality solutions on all the instances.

The impact on solutions of a number of problem characteristics was analyzed, in particular related to the availability periods of items and bins. We also explored the interest of explicitly integrating the time-dependency of items, bins, and their related decisions to the bin packing formulations. The results of the computational experiments confirm the importance of explicitly considering the time dimensions of the problem, as one observes significant reductions in operational costs provided by the multi-period formulation compared to a myopic approach.

Given the fast computational times of the proposed heuristics, it appears that they can be used jointly to address a problem instance. That is, one could run the four heuristics for the multi-period problem and select the best result in a very short time. Such an approach would both potentially improve the solution and hedge against the risk of making the wrong choice of algorithm, while remaining computationally efficient. It also appears quite clearly that these algorithms can be used within more sophisticated solution methods algorithms for improved search performance, as well as algorithmic components of solution methods for more complex problem settings, e.g., stochastic versions. Our experiments actually showed that, using the solution provided by the heuristic as initial

upper bound (and, thus, initial cut-off threshold for the pruning of the tree) improved significantly the performance of the exact solution method of the commercial solver.

To summarize, the numerical results thus show the interest of the new problem settings and formulations we introduce, the high-performance and robustness, in terms of solution quality and computational efficiency, of the constructive heuristics we propose, the interest of using these heuristics as components of more sophisticated solution methods, as well as the potential benefits of using the new models in practical applications.

Several research avenues may be now contemplated. Applying the methodology to more practical settings and ranges of problem characteristics would raise new issues and provide the opportunity to enhance the models and methods. Richer problem settings should also be investigated, e.g., by increasing the packing dimensions and introducing particular loading rules. Extending this research to investigate the various sources of uncertainty that may be related to items and bins and propose appropriate stochastic-optimization models is also a challenging but fascinating research avenue. Last but not least, one should start the development of more sophisticated meta- and matheuristic algorithms for these problems and models. We hope to report on some of these issues in the near future.

# Acknowledgements

While working on the project, the first author was Adjunct Professor, Department of Computer Science and Operations Research, Université de Montréa, while the second was Postdoctoral Researcher, School of Management, UQAM, and CIRRELT. We gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery, Acceleration Grant and Engage programs. We also gratefully acknowledge the support of Fonds de recherche Nature et Technologies du Québec through their team research project program. Thanks are also due to Calcul Quebec and Compute Canada for providing the authors access to their high-performance computing infrastructure.

# References

Álvarez-Miranda, E. and Pereira, J. (2019). On the complexity of assembly line balancing problems. *Computers & Operations Research*, 108:182–186.

Alves, C. and de Carvalho, J. V. (2007). Accelerating column generation for variable sized bin-packing problems. *European Journal of Operational Research*, 183(3):1333 – 1352.

Aydın, N., İbrahim Muter, and Ş. İlker Birbil (2020). Multi-objective temporal bin packing problem: An application in cloud computing. *Computers & Operations Research*, 121:104959.

Baldi, M. M., Crainic, T. G., Perboli, G., and Tadei, R. (2012). The generalized bin packing problem. *Transportation Research Part E*, 48:1205–1220.

Baldi, M. M., Crainic, T. G., Perboli, G., and Tadei, R. (2014). Branch-and-price and beam search algorithms for the variable cost and size bin packing problem with optional items. *Annals of Operations Research*, 222:125–141.

Baldi, M. M., Manerba, D., Perboli, G., and Tadei, R. (2019). A generalized bin packing problem for parcel delivery in last-mile logistics. *European Journal of Operational Research*, 274:990–999.

Battaïa, O. and Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal Production Economics*, 142:259–277.

Brandão, F. and Pedroso, J. P. (2016). Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, 69:56 – 67.

Casazza, M. and Ceselli, A. (2016). Exactly solving packing problems with fragmentation. *Computers & Operations Research*, 75:202 – 213.

Correia, I., Gouveia, L., and da Gama, F. S. (2008). Solving the variable size bin packing problem with discretized formulations. *Computers and Operations Research*, 35(6):2103 – 2113. Part Special Issue: OR Applications in the Military and in Counter-Terrorism.

Crainic, T. G., Gobbato, L., Perboli, G., and Rei, W. (2016). Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic. *European Journal of Operational Research*, 253(2):404 – 417.

Crainic, T. G., Perboli, G., Rei, W., and Tadei, R. (2011). Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers and Operations Research*, 38:1474–1482.

Crainic, T. G., Perboli, G., and Tadei, R. (2012). Recent advances in multi-dimensional packing problems. *In C. Volosencu (Ed.), New Technologies-trends, innovations and research*, pages 91–110.

Dell'Amico, M., Furini, F., and Iori, M. (2020). A branch-and-price algorithm for the temporal bin packing problem. *Computers & Operations Research*, 114:104825.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44:145–159.

Dyckhoff, H., Stheithauer, G., and Terno, J. (1997). Cutting and packing. In M. Dell'Amico, F. M. and Martello, S., editors, *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, Chichester, U.K.

Fanjul-Peyro, L., Perea, F., and Ruiz, R. (2017). Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research*, 260(2):482 – 493.

Fomeni, F. D., Rei, W., and Crainic, T. G. (2020). Data set related to the paper: "Multi-period bin packing model and effective constructive heuristics for a corridor-based logistics capacity planning". http://dx.doi.org/10.17632/z3rv8dhm37.2. Mendeley Data.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. W. H. Freeman.

GNU (2018). GCC, the GNU Compiler Collection version 4.8.1. https://gcc.gnu.org/.

Huang, M., Cui, Y., Yang, S., and Wang, X. (2013). Fourth party logistics routing problem with fuzzy duration time. *International Journal of Production Economics*, 145:107–116.

IBM (2016). ILOG CPLEX Optimization Studio. www-4201.ibm.com/software/commerce/optimization/cplex-optimizer/.

Kang, J. and Park, S. (2003). Algorithms for the variable sized bin packing problem. *European Journal of Operational Research*, 147(2):365 – 372. Fuzzy Sets in Scheduling and Planning.

Korte, B. and Vygen, J. (2006). *Combinatorial Optimization: Theory and Algorithms. Algorithms and Combinatorics*. Springer-Verlag, Heidelberg, Germany.

Kravchenko, S. A. and Werner, F. (2011). Parallel machine problems with equal processing times: a survey. *Journal of Scheduling*, 14:435–444.

Leao, A. A., Toledo, F. M., Oliveira, J. F., Carravilla, M. A., and Álvarez-Valdés, R. (2020). Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803 – 822.

Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P. (1977). Complexity of machine scheduling problems. In Hammer, P., Johnson, E., Korte, B., and Nemhauser, G., editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 343 – 362. Elsevier.

Martello, S. and Toth, P. (1990). *Knapsack Problems - Algorithms and ComputImplementations*. John Wiley & Sons, Chichester, UK.

Martinovic, J., Strasdat, N., and Selch, M. (2021). Compact integer linear programming formulations for the temporal bin packing problem with fire-ups. *Computers & Operations Research*, page 105288.

Monaci, M. (2002). Algorithms for packing and scheduling problems. *PhD Thesis*, University of Bologna.

Monczka, R., Handfield, R., Giunipero, L., and Patterson, J. (2008). Purchasing and supply chain management. Cengage Learning.

Özceylan, E., Kalayci, C. B., Güngör, A., and Gupta, S. M. (2019). Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, 57(15-16):4805–4827.

Perboli, G., Gobbato, G., and Perfetti, F. (2014). Packing problems in transportation and supply chain: New problems and trends. *Procedia–Social and Behavioral Sciences*, 111:672–681.

Pisinger, D. and Sigurd, M. (2005). The two-dimensional bin packing problem with variable bin sizes and costs. *Discrete Optimization*, 2(2):154 – 167.

Wäscher, G., Haußner, H., and Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109–1130.

Wee, T. S. and Magazine, M. J. (1982). Assembly line balancing as generalized bin packing. *Operations Research Letters*, 1(2):56–58.

Yilmaz, O. and Savasaneril, S. (2012). Collaboration among small shippers in a transportation market. *European Journal of Operational Research*, 218:408–415.

# Appendix - Pseudo-code of Algorithm HS3

**Algorithm 6** Heuristic HS3 for the single-period

*Input:* Sets of items $\mathcal{I}$ and bins $\mathcal{J}$;
*Output:* Sets $\mathcal{A}$ of items assigned to bins and $\mathcal{S}$ of selected bins;
Initialize $\mathcal{S} = \emptyset$, $\mathcal{A} = \emptyset$;
**for** all $j \in \mathcal{J}$ **do**
  $c_j = f_j$;
  Sort the items in $\mathcal{I}$ in increasing order of $\bar{a}_{ij}$;
  **for** all $i \in$ sorted $\mathcal{I}$ **do**
    **if** $v_i \leq V_j$ and $a_{ij} < p_i^*$ **then** Load $i$ into $j$; $c_j := c_j + a_{ij}$; $V_j := V_j - v_i$;
    **end if**
  **end for**
**end for**
Sort the bins in $\mathcal{J}$ in increasing order of $c_j$;
Re-initialize $V_j$ to the initial capacity of the bin, for all $j \in \mathcal{J}$;
**for** all $j \in$ sorted $\mathcal{J}$ **do**
  Sort the items in $\mathcal{I}$ in increasing order of $\bar{a}_{ij}$;
  **for** all $i \in$ sorted $\mathcal{I}$ **do**
    **if** $v_i \leq V_j$ and $a_{ij} < p_i$ **then** Load $i$ into $j$; $\mathcal{A} := \mathcal{A} \cup \{i\}$, $V_j := V_j - v_i$;
    **end if**
  **end for**
  **if** $j$ is non-empty **then** $\mathcal{S} := \mathcal{S} \cup \{j\}$;
  **end if**
**end for**
**for** all bin $j \in \mathcal{S}$ with residual capacity $V_j^*$ **do**
  Sort the items in $\mathcal{I} \setminus \mathcal{A}$ in increasing order of $\bar{a}_{ij}$;
  **for** $i \in$ sorted $\mathcal{I} \setminus \mathcal{A}$ **do**
    **if** $V_j^* - v_i \geq 0$ **then** Load $i$ into $j$; $\mathcal{A} := \mathcal{A} \cup \{i\}$, $V_j^* := V_j^* - v_i$;
    **end if**
  **end for**
**end for**
**for** $i \in \mathcal{A}$ assigned to a bin $j$ **do**
  **for** $i' \in \mathcal{I} \setminus \mathcal{A}$ **do**
    **if** $(V_j^* - v_{i'} + v_i \geq 0)$ and $(a_{ij} > a_{i'j})$ **then**
      Replace item $i$ with item $i'$ in bin $j$;
      Update the sets of assigned and unassigned items appropriately;
      $V_j^* := V_j^* - v_{i'} + v_i$;
    **end if**
  **end for**
**end for**
Assign any still unassigned item to a spot-market bin.

Table 4: Performance for on Set 2 instances

| | | HS1 | | HS2 | | HS3 | | BnC | |
|---|---|---|---|---|---|---|---|---|---|
| | $\|\mathcal{I}\|$ | Gap | Time | Gap | Time | Gap | Time | Gap | Time |
| Type 1 | 100 | 2.52 | 0.11 | 7.89 | 0.01 | 7.04 | 0.01 | 0.00 | 2.92 |
| | 200 | 4.44 | 0.70 | 8.68 | 0.04 | 9.15 | 0.06 | 0.01 | 23.58 |
| | 300 | 4.11 | 1.56 | 8.96 | 0.12 | 9.58 | 0.16 | 0.01 | 51.05 |
| | 500 | 4.51 | 4.43 | 9.12 | 0.36 | 10.16 | 0.45 | 0.01 | 169.04 |
| | 750 | 3.69 | 10.23 | 9.65 | 0.93 | 10.11 | 1.05 | 0.01 | 356.58 |
| | 1000 | 3.10 | 17.20 | 9.31 | 1.35 | 9.59 | 2.16 | 0.01 | 908.36 |
| Type 2 | 100 | 0.01 | 0.10 | 0.01 | 0.01 | 0.01 | 0.01 | 0.00 | 0.21 |
| | 200 | 0.29 | 0.67 | 0.29 | 0.02 | 0.29 | 0.04 | 0.01 | 0.96 |
| | 300 | 0.22 | 2.05 | 0.19 | 0.06 | 0.19 | 0.09 | 0.01 | 2.69 |
| | 500 | 0.35 | 5.67 | 0.38 | 0.16 | 0.38 | 0.23 | 0.01 | 53.53 |
| | 750 | 0.41 | 12.45 | 0.41 | 0.33 | 0.41 | 0.49 | 0.01 | 66.68 |
| | 1000 | 0.39 | 20.99 | 0.38 | 0.58 | 0.38 | 0.86 | 0.01 | 47.63 |
| Type 3 | 100 | 0.00 | 0.18 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.20 |
| | 200 | 0.00 | 1.08 | 0.00 | 0.03 | 0.00 | 0.05 | 0.00 | 0.32 |
| | 300 | 0.00 | 3.44 | 0.00 | 0.08 | 0.00 | 0.12 | 0.00 | 0.46 |
| | 500 | 0.00 | 9.77 | 0.00 | 0.22 | 0.00 | 0.32 | 0.00 | 0.81 |
| | 750 | 0.00 | 20.95 | 0.00 | 0.47 | 0.00 | 0.68 | 0.00 | 1.20 |
| | 1000 | 0.00 | 36.22 | 0.00 | 0.81 | 0.00 | 1.21 | 0.00 | 1.66 |
| Type 4 | 100 | 3.17 | 0.11 | 13.41 | 0.01 | 13.41 | 0.01 | 0.00 | 0.85 |
| | 200 | 4.22 | 0.75 | 14.78 | 0.03 | 14.78 | 0.04 | 0.00 | 1.43 |
| | 300 | 2.99 | 1.60 | 14.87 | 0.08 | 14.87 | 0.12 | 0.01 | 1.36 |
| | 500 | 3.52 | 4.52 | 14.68 | 0.28 | 14.68 | 0.44 | 0.01 | 2.86 |
| | 750 | 3.07 | 9.29 | 14.43 | 0.45 | 14.43 | 0.70 | 0.01 | 5.43 |
| | 1000 | 2.68 | 14.95 | 13.61 | 0.69 | 13.61 | 1.07 | 0.01 | 8.66 |
| Type 5 | 100 | 0.61 | 0.10 | 0.70 | 0.01 | 0.70 | 0.01 | 0.00 | 0.25 |
| | 200 | 1.14 | 0.95 | 1.39 | 0.02 | 1.39 | 0.03 | 0.01 | 0.92 |
| | 300 | 1.02 | 3.34 | 1.21 | 0.05 | 1.21 | 0.08 | 0.01 | 1.27 |
| | 500 | 1.04 | 8.73 | 1.31 | 0.15 | 1.31 | 0.22 | 0.01 | 3.26 |
| | 750 | 1.04 | 20.65 | 1.24 | 0.33 | 1.24 | 0.48 | 0.00 | 6.88 |
| | 1000 | 0.97 | 35.76 | 1.09 | 0.57 | 1.09 | 0.84 | 0.01 | 11.65 |
| Type 6 | 100 | 0.00 | 0.17 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.19 |
| | 200 | 0.00 | 1.07 | 0.00 | 0.03 | 0.00 | 0.05 | 0.00 | 0.31 |
| | 300 | 0.00 | 3.24 | 0.00 | 0.07 | 0.00 | 0.12 | 0.00 | 0.48 |
| | 500 | 0.00 | 8.92 | 0.00 | 0.21 | 0.00 | 0.31 | 0.00 | 0.82 |
| | 750 | 0.00 | 19.92 | 0.00 | 0.46 | 0.00 | 0.66 | 0.00 | 1.33 |
| | 1000 | 0.00 | 34.85 | 0.00 | 0.80 | 0.00 | 1.17 | 0.00 | 1.65 |

Table 5: Performance for 1/V/V/S/N[I2B] on Set 1 instances

| | | HM1 | | HM2 | | HM3 | | HM4 | | BnC | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\|\mathcal{I}\|$ | Gap | Time | Gap | Time | Gap | Time | Gap | Time | $Gap^{10}$ | Gap | Time | #Sol. | #Feas. |
| Type 1 | 100 | 15.82 | 0.01 | 12.19 | 0.02 | 12.90 | 0.03 | 7.45 | 0.05 | 11.67 | 0.00 | 16.12 | 10 | 10 |
| | 200 | 16.24 | 0.06 | 15.71 | 0.14 | 16.05 | 0.23 | 7.07 | 0.23 | 13.03 | 0.06 | 1,493.74 | 8 | 10 |
| | 300 | 12.32 | 0.13 | 14.30 | 0.37 | 14.47 | 0.69 | 8.07 | 0.51 | 12.66 | 0.09 | 3,035.26 | 8 | 10 |
| | 500 | 7.89 | 0.39 | 13.24 | 1.19 | 14.72 | 2.18 | 8.84 | 1.21 | 12.20 | 0.12 | 2,650.34 | 6 | 10 |
| | 750 | 4.27 | 1.05 | 8.19 | 3.52 | 9.68 | 7.58 | 4.36 | 2.96 | 11.81 | 0.15 | 2,917.40 | 6 | 10 |
| | 1000 | 1.06 | 2.50 | 5.53 | 8.11 | 7.31 | 15.84 | 0.94 | 6.90 | 12.67 | 0.11 | 3,600.21 | 6 | 10 |
| Type 2 | 100 | 16.81 | 0.02 | 15.73 | 0.05 | 14.76 | 0.07 | 8.76 | 0.11 | 13.49 | 0.01 | 103.48 | 10 | 10 |
| | 200 | 14.56 | 0.06 | 14.54 | 0.19 | 15.25 | 0.26 | 7.23 | 0.24 | 12.05 | 0.02 | 1,072.54 | 9 | 10 |
| | 300 | 10.77 | 0.13 | 11.12 | 0.54 | 11.36 | 0.85 | 5.71 | 0.81 | 14.47 | 0.47 | 3,600.05 | 2 | 9 |
| | 500 | 11.47 | 0.54 | 12.50 | 2.99 | 12.89 | 4.98 | 5.76 | 3.18 | 14.00 | 0.73 | 3,600.06 | 0 | 9 |
| | 750 | 5.58 | 1.34 | 7.12 | 8.18 | 7.76 | 13.83 | 1.90 | 6.97 | 13.34 | 0.33 | 3,600.19 | 0 | 6 |
| | 1000 | 1.45 | 2.67 | 3.17 | 15.35 | 3.98 | 25.81 | 1.16 | 13.33 | 14.49 | 0.52 | 3,600.33 | 0 | 3 |
| Type 3 | 100 | 14.40 | 0.03 | 14.86 | 0.09 | 16.12 | 0.11 | 7.94 | 0.18 | 16.05 | 0.01 | 124.89 | 10 | 10 |
| | 200 | 17.04 | 0.09 | 16.09 | 0.37 | 17.17 | 0.56 | 8.31 | 0.64 | 14.21 | 0.20 | 1,991.09 | 7 | 10 |
| | 300 | 14.12 | 0.19 | 13.68 | 0.94 | 13.63 | 1.41 | 7.05 | 1.38 | 16.50 | 0.56 | 3,600.01 | 0 | 10 |
| | 500 | 11.47 | 0.54 | 12.50 | 2.99 | 12.89 | 4.89 | 5.76 | 2.98 | 14.00 | 0.73 | 3,600.02 | 0 | 9 |
| | 750 | 5.44 | 1.34 | 6.95 | 8.27 | 7.59 | 13.69 | 1.75 | 5.61 | 13.36 | 0.33 | 3,600.29 | 0 | 6 |
| | 1000 | 0.17 | 2.68 | 1.87 | 16.41 | 2.67 | 25.98 | 1.41 | 15.23 | 17.05 | 0.47 | 3,600.38 | 0 | 2 |
| Type 4 | 100 | 15.64 | 0.02 | 12.65 | 0.04 | 13.00 | 0.07 | 8.11 | 0.09 | 14.33 | 0.01 | 90.11 | 10 | 10 |
| | 200 | 14.12 | 0.07 | 15.44 | 0.22 | 14.70 | 0.38 | 7.70 | 0.37 | 21.47 | 0.04 | 2,770.56 | 8 | 10 |
| | 300 | 13.54 | 0.18 | 15.27 | 0.62 | 15.08 | 1.12 | 7.60 | 0.82 | 24.05 | 0.09 | 3,437.43 | 3 | 10 |
| | 500 | 10.03 | 0.59 | 12.17 | 2.28 | 12.52 | 4.07 | 6.27 | 2.43 | 14.37 | 0.24 | 3,600.03 | 0 | 9 |
| | 750 | 5.18 | 1.60 | 8.13 | 5.61 | 9.21 | 10.84 | 3.27 | 4.96 | 12.66 | 0.30 | 3,600.21 | 0 | 6 |
| | 1000 | 4.34 | 3.12 | 7.80 | 10.21 | 10.02 | 21.36 | 3.69 | 8.45 | 15.33 | 0.38 | 3,600.13 | 0 | 6 |
| Type 5 | 100 | 11.13 | 0.03 | 11.26 | 0.09 | 12.58 | 0.17 | 11.38 | 0.26 | 17.39 | 2.49 | 2,774.82 | 3 | 10 |
| | 200 | 6.32 | 0.14 | 7.28 | 0.48 | 7.51 | 0.94 | 5.89 | 1.08 | 20.76 | 3.51 | 3,600.02 | 0 | 10 |
| | 300 | 7.88 | 0.24 | 8.76 | 0.91 | 8.40 | 1.83 | 7.39 | 1.63 | 12.94 | 2.16 | 3,600.06 | 0 | 10 |
| | 500 | 6.10 | 0.87 | 7.86 | 3.83 | 7.37 | 7.14 | 6.15 | 5.38 | 8.82 | 1.47 | 3,600.04 | 0 | 8 |
| | 750 | 5.73 | 2.38 | 6.03 | 11.90 | 7.00 | 23.47 | 6.13 | 13.40 | 5.63 | 1.86 | 3,600.06 | 0 | 8 |
| | 1000 | 6.69 | 3.78 | 7.08 | 20.97 | 7.98 | 42.14 | 6.74 | 16.40 | 18.39 | 1.18 | 3,600.06 | 0 | 10 |

Table 6: Performance for 1/V/V/S/N[I2B] on Set 2 instances

| | | HM1 | | HM2 | | HM3 | | HM4 | | BnC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\|\mathcal{I}\|$ | Gap | Time | Gap | Time | Gap | Time | Gap | Time | Gap | Time | #Sol. | #Feas. |
| Type 1 | 100 | 2.01 | 0.06 | 3.68 | 0.01 | 3.96 | 0.02 | 3.96 | 0.30 | 0.01 | 2.74 | 10 | 10 |
| | 200 | 3.42 | 0.20 | 5.93 | 0.08 | 6.20 | 0.13 | 5.57 | 1.74 | 0.01 | 13.69 | 10 | 10 |
| | 300 | 3.90 | 0.42 | 6.17 | 0.23 | 7.76 | 0.28 | 6.57 | 3.32 | 0.01 | 66.94 | 10 | 10 |
| | 500 | 3.16 | 1.42 | 6.05 | 0.83 | 7.53 | 1.11 | 6.05 | 9.92 | 0.01 | 441.73 | 10 | 10 |
| | 750 | 3.69 | 3.42 | 5.89 | 2.88 | 7.68 | 3.19 | 6.24 | 22.31 | 0.02 | 1,530.40 | 9 | 10 |
| | 1000 | 2.31 | 7.61 | 4.68 | 6.02 | 6.60 | 6.66 | 5.32 | 36.61 | 0.02 | 2,849.11 | 8 | 8 |
| Type 2 | 100 | 0.00 | 0.05 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.45 | 0.00 | 0.11 | 10 | 10 |
| | 200 | 0.00 | 0.17 | 0.00 | 0.04 | 0.00 | 0.06 | 0.00 | 1.72 | 0.00 | 0.30 | 10 | 10 |
| | 300 | 0.04 | 0.41 | 0.04 | 0.09 | 0.04 | 0.14 | 0.04 | 3.96 | 0.00 | 0.63 | 10 | 10 |
| | 500 | 0.14 | 1.20 | 0.14 | 0.22 | 0.14 | 0.36 | 0.14 | 11.27 | 0.01 | 4.67 | 10 | 10 |
| | 750 | 0.15 | 2.72 | 0.15 | 0.46 | 0.15 | 0.78 | 0.15 | 25.09 | 0.01 | 8.28 | 10 | 10 |
| | 1000 | 0.18 | 5.11 | 0.18 | 0.76 | 0.18 | 1.25 | 0.18 | 42.57 | 0.01 | 225.17 | 10 | 10 |
| Type 3 | 100 | 0.00 | 0.08 | 0.00 | 0.02 | 0.00 | 0.03 | 0.00 | 0.80 | 0.00 | 0.12 | 10 | 10 |
| | 200 | 0.00 | 0.31 | 0.00 | 0.06 | 0.00 | 0.08 | 0.00 | 3.36 | 0.00 | 0.19 | 10 | 10 |
| | 300 | 0.00 | 0.72 | 0.00 | 0.12 | 0.00 | 0.18 | 0.00 | 7.71 | 0.00 | 0.26 | 10 | 10 |
| | 500 | 0.00 | 1.90 | 0.00 | 0.28 | 0.00 | 0.54 | 0.00 | 18.66 | 0.00 | 0.40 | 10 | 10 |
| | 750 | 0.00 | 4.50 | 0.00 | 0.61 | 0.00 | 1.05 | 0.00 | 41.09 | 0.00 | 0.66 | 10 | 10 |
| | 1000 | 0.00 | 8.17 | 0.00 | 1.03 | 0.00 | 1.73 | 0.00 | 71.41 | 0.00 | 0.86 | 10 | 10 |
| Type 4 | 100 | 0.65 | 0.05 | 1.32 | 0.02 | 1.42 | 0.02 | 1.42 | 0.32 | 0.01 | 24.07 | 10 | 10 |
| | 200 | 2.32 | 0.20 | 3.63 | 0.10 | 4.38 | 0.10 | 4.41 | 1.76 | 0.01 | 148.40 | 10 | 10 |
| | 300 | 3.09 | 0.37 | 5.56 | 0.27 | 6.19 | 0.31 | 6.14 | 3.94 | 0.01 | 142.31 | 10 | 10 |
| | 500 | 3.02 | 1.27 | 5.97 | 1.04 | 6.75 | 1.04 | 5.88 | 10.66 | 0.02 | 1,728.06 | 10 | 10 |
| | 750 | 2.31 | 3.29 | 4.64 | 3.14 | 5.92 | 3.28 | 4.81 | 21.27 | 0.05 | 2,377.78 | 6 | 8 |
| | 1000 | 2.50 | 6.03 | 4.50 | 6.60 | 5.88 | 8.44 | 5.01 | 41.56 | 0.07 | 3,462.01 | 4 | 8 |
| Type 5 | 100 | 0.00 | 0.04 | 0.00 | 0.02 | 0.00 | 0.03 | 0.00 | 0.54 | 0.00 | 0.07 | 10 | 10 |
| | 200 | 0.00 | 0.14 | 0.00 | 0.06 | 0.00 | 0.08 | 0.00 | 2.00 | 0.00 | 0.14 | 10 | 10 |
| | 300 | 0.00 | 0.34 | 0.00 | 0.11 | 0.00 | 0.16 | 0.00 | 4.84 | 0.00 | 0.24 | 10 | 10 |
| | 500 | 0.13 | 1.19 | 0.13 | 0.22 | 0.13 | 0.35 | 0.13 | 11.51 | 0.01 | 4.15 | 10 | 10 |
| | 750 | 0.15 | 2.73 | 0.15 | 0.46 | 0.15 | 0.74 | 0.15 | 24.51 | 0.01 | 8.19 | 10 | 10 |
| | 1000 | 0.18 | 5.11 | 0.18 | 0.76 | 0.18 | 1.22 | 0.18 | 40.39 | 0.01 | 222.81 | 10 | 10 |
| Type 6 | 100 | 0.00 | 0.06 | 0.00 | 0.03 | 0.00 | 0.04 | 0.00 | 0.98 | 0.00 | 0.07 | 10 | 10 |
| | 200 | 0.00 | 0.25 | 0.00 | 0.08 | 0.00 | 0.11 | 0.00 | 3.63 | 0.00 | 0.13 | 10 | 10 |
| | 300 | 0.00 | 0.58 | 0.00 | 0.15 | 0.00 | 0.21 | 0.00 | 8.36 | 0.00 | 0.18 | 10 | 10 |
| | 500 | 0.00 | 1.55 | 0.00 | 0.34 | 0.00 | 0.50 | 0.00 | 20.76 | 0.00 | 0.26 | 10 | 10 |
| | 750 | 0.00 | 3.66 | 0.00 | 0.70 | 0.00 | 1.07 | 0.00 | 46.58 | 0.00 | 0.42 | 10 | 10 |
| | 1000 | 0.03 | 6.06 | 0.03 | 1.12 | 0.03 | 1.77 | 0.03 | 74.97 | 0.00 | 4.00 | 10 | 10 |

Table 7: Bin and item availability-window impact analysis

| TW Width | $\|\mathcal{I}\|$ | $\|\mathcal{J}\|$ | HM4 | CPLEX | Gap (%) |
|---|---|---|---|---|---|
| | 100 | 90 | 0.01 | 15.35 | 0.18 |
| | 200 | 87 | 0.04 | 180.04 | 0.68 |
| Items: [0,2] | 300 | 86 | 0.09 | 308.36 | 4.98 |
| Bins: [0,2] | 500 | 83 | 0.27 | 829 | 6.19 |
| | 750 | 91 | 0.94 | 37.03 | 6.94 |
| | 1000 | 90 | 1.75 | 26.46 | 8.29 |
| | 100 | 90 | 0.01 | 1.39 | 0.14 |
| | 200 | 87 | 0.03 | 55.96 | 2.13 |
| Items: [0,2] | 300 | 86 | 0.11 | 84.22 | 3.14 |
| Bins: [0,0] | 500 | 83 | 0.43 | 11.33 | 5.13 |
| | 750 | 91 | 1.33 | 5.39 | 5.14 |
| | 1000 | 90 | 2.85 | 6.78 | 5.78 |
| | 100 | 90 | 0.01 | 0.4 | 0.01 |
| | 200 | 87 | 0.03 | 14.73 | 0.12 |
| Items: [0,0] | 300 | 86 | 0.06 | 208.25 | 0.41 |
| Bins: [0,2] | 500 | 83 | 0.16 | 240.00 | 2.35 |
| | 750 | 91 | 0.53 | 202.99 | 2.73 |
| | 1000 | 90 | 1.16 | 544.83 | 3.53 |
| | 100 | 90 | 0.01 | 0.03 | 0.01 |
| | 200 | 87 | 0.02 | 0.04 | 0.08 |
| Items: [0,0] | 300 | 86 | 0.04 | 0.05 | 0.24 |
| Bins: [0,0] | 500 | 83 | 0.17 | 0.1 | 1.4 |
| | 750 | 91 | 0.71 | 0.17 | 3.33 |
| | 1000 | 90 | 1.66 | 0.25 | 2.46 |

Table 8: Average Improvement of our heuristics with respect to the myopic approach

| | | Set 1 | | | | Set 2 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\|\mathcal{I}\|$ | HM1 | HM2 | HM3 | HM4 | HM1 | HM2 | HM3 | HM4 |
| Type 1 | 100 | 14.98 | 17.61 | 17.07 | 21.09 | 73.84 | 73.39 | 73.32 | 73.32 |
| | 200 | 24.17 | 24.45 | 24.20 | 30.02 | 62.21 | 61.29 | 61.19 | 61.42 |
| | 300 | 22.82 | 21.34 | 21.19 | 25.47 | 51.78 | 50.76 | 50.01 | 50.56 |
| | 500 | 17.60 | 13.27 | 12.09 | 16.53 | 37.59 | 35.83 | 34.94 | 35.84 |
| | 750 | 14.68 | 11.31 | 10.00 | 14.39 | 27.55 | 26.02 | 24.76 | 25.77 |
| | 1000 | 13.95 | 10.10 | 8.56 | 13.92 | 23.09 | 21.31 | 19.87 | 20.82 |
| Type 2 | 100 | 41.52 | 42.05 | 42.52 | 45.48 | 77.82 | 77.82 | 77.82 | 77.82 |
| | 200 | 23.98 | 23.99 | 23.46 | 28.86 | 69.96 | 69.96 | 69.96 | 69.96 |
| | 300 | 20.19 | 19.90 | 19.74 | 23.79 | 61.81 | 61.81 | 61.81 | 61.81 |
| | 500 | 17.62 | 16.86 | 16.56 | 21.82 | 49.27 | 49.27 | 49.27 | 49.27 |
| | 750 | 19.95 | 18.78 | 18.31 | 22.75 | 38.24 | 38.24 | 38.24 | 38.24 |
| | 1000 | 17.53 | 16.12 | 15.43 | 18.79 | 31.14 | 31.14 | 31.14 | 31.14 |
| Type 3 | 100 | 55.90 | 55.76 | 55.27 | 58.38 | 81.64 | 81.64 | 81.64 | 81.64 |
| | 200 | 36.86 | 37.35 | 36.76 | 41.52 | 75.97 | 75.97 | 75.97 | 75.97 |
| | 300 | 25.38 | 25.66 | 25.69 | 29.99 | 69.44 | 69.44 | 69.44 | 69.44 |
| | 500 | 17.62 | 16.86 | 16.56 | 21.82 | 57.31 | 57.31 | 57.31 | 57.31 |
| | 750 | 19.95 | 18.78 | 18.31 | 22.75 | 47.55 | 47.55 | 47.55 | 47.55 |
| | 1000 | 17.53 | 16.12 | 15.43 | 18.79 | 40.89 | 40.89 | 40.89 | 40.89 |
| Type 4 | 100 | 16.89 | 19.08 | 18.79 | 22.27 | 72.40 | 72.21 | 72.18 | 72.18 |
| | 200 | 28.06 | 27.25 | 27.66 | 32.07 | 64.37 | 63.91 | 63.65 | 63.64 |
| | 300 | 25.25 | 24.12 | 24.26 | 29.16 | 55.09 | 54.00 | 53.73 | 53.75 |
| | 500 | 18.04 | 16.45 | 16.17 | 20.79 | 41.03 | 39.33 | 38.88 | 39.38 |
| | 750 | 14.35 | 11.87 | 11.03 | 15.83 | 30.58 | 29.01 | 28.14 | 28.90 |
| | 1000 | 13.61 | 10.70 | 8.85 | 14.10 | 24.00 | 22.53 | 21.52 | 22.16 |
| Type 5 | 100 | 37.56 | 37.49 | 36.78 | 37.42 | 78.16 | 78.16 | 78.16 | 78.16 |
| | 200 | 27.60 | 26.96 | 26.78 | 27.91 | 71.90 | 71.90 | 71.90 | 71.90 |
| | 300 | 20.13 | 19.47 | 19.74 | 20.49 | 64.81 | 64.81 | 64.81 | 64.81 |
| | 500 | 14.53 | 13.10 | 13.51 | 14.48 | 49.75 | 49.75 | 49.75 | 49.75 |
| | 750 | 12.58 | 12.35 | 11.52 | 12.25 | 38.24 | 38.24 | 38.24 | 38.24 |
| | 1000 | 11.74 | 11.42 | 10.67 | 11.71 | 31.14 | 31.14 | 31.14 | 31.14 |
| Type 6 | 100 | | | | | 82.06 | 82.06 | 82.06 | 82.06 |
| | 200 | | | | | 77.58 | 77.58 | 77.58 | 77.58 |
| | 300 | | | | | 72.50 | 72.50 | 72.50 | 72.50 |
| | 500 | | | | | 60.75 | 60.75 | 60.75 | 60.75 |
| | 750 | | | | | 50.80 | 50.80 | 50.80 | 50.80 |
| | 1000 | | | | | 41.82 | 41.82 | 41.82 | 41.82 |