The 2-D Bin Packing Problem with Multiple Levels of Prioritization: A Spatial Optimization Perspective

William K. Kirschenman[a,*], H. Sebastian Heese[b], Michael G. Kay[a], Russell E. King[a], Brandon M. McConnell[a,c]

[a] *Department of Industrial and Systems Engineering, North Carolina State University, Campus Box 7906, Raleigh, 27695-7906, NC, United States*
[b] *Poole College of Management, North Carolina State University, 2300 Nelson Hall, Raleigh, 27695-7229, NC, United States*
[c] *Center for Additive Manufacturing and Logistics (CAMAL), North Carolina State University, 915 Partners Way, Box 7906, Raleigh, 27695-7906, NC, United States*

Abstract

This paper integrates two-dimensional bin packing with facility layout concepts to address scenarios in which items must not only fit within a confined space but also be arranged according to spatial priorities. We embed a prioritization matrix into the bin packing framework, enabling items to be clustered with one another or pulled toward certain bin access points based on assigned priority weights. Unlike traditional bin packing, which focuses on space utilization alone, our approach balances proximity to bin access points and adjacency among functionally related items, extending the utility of bin packing to applications requiring more nuanced layout preferences.

We introduce a single mixed-integer linear programming (MILP) model and a complementary sliding window decomposition method that scales effectively to larger problem instances. Numerical experiments illustrate that this decomposition approach consistently outperforms a direct MILP solve with a commercial solver in both runtime and solution quality. This computational study underscores the flexibility and effectiveness of embedding multi-level priorities into bin packing.

Keywords: Integer programming, Facilities planning and design, Packing, Combinatorial optimization, Logistics

1. Introduction

The problem of efficiently arranging objects within a confined space is a fundamental challenge in operations research, commonly studied as a bin packing problem. Classical formulations of bin packing primarily focus on minimizing either the number of bins used or the amount of unused space within each bin. However, many real-world applications impose additional requirements beyond spatial efficiency. In certain contexts, the placement of items relative to

*Corresponding author at: Department of Industrial and Systems Engineering, North Carolina State University, Campus Box 7906, Raleigh, NC 27695-7906, United States. Email: wkkirsch@ncsu.edu

Email addresses: `wkkirsch@ncsu.edu` (William K. Kirschenman), `hsheese@ncsu.edu` (H. Sebastian Heese), `kay@ncsu.edu` (Michael G. Kay), `king@ncsu.edu` (Russell E. King), `mcconnell@ncsu.edu` (Brandon M. McConnell)

other objects or specific reference points within the bin is equally important. In such settings, classical bin packing's emphasis on space utilization alone becomes insufficient, as operational priorities must also be addressed. These priorities may arise from access sequencing constraints, the need to maintain proximity between functionally related items, or requirements to position high-priority items near designated locations, such as loading doors or off-loading points.

A related stream of research in facility layout planning addresses the optimal arrangement of departments, machines, or storage units within a defined space to minimize transportation costs, congestion, or material handling inefficiencies. These problems often consider adjacency-based cost structures, where minimizing weighted distances between specific objects is a primary objective. By contrast, two-dimensional bin packing formulations primarily focus on ensuring feasibility under geometric constraints, such as non-overlapping placement and orientation restrictions. This research integrates elements of both domains by incorporating a prioritization-weighted cost function into a classical bin packing framework. Specifically, we introduce a prioritization-weight matrix that captures item-to-item and item-to-bin access point placement preferences, thereby extending the bin packing problem to include spatial considerations traditionally found in facility layout optimization.

To illustrate the core challenge and the impact of our approach, consider a small example involving six items belonging to three distinct groups, each with varying priorities for access point proximity (detailed in Table 1). These items are to be placed within a rectangular bin of 700 length and 350 width. Items within a group should ideally be kept close together (group cohesion), while high-priority items should be near a designated bin access point (here, the center-left edge of the bin).

Table 1: Example instance illustrating groups and priorities.

Item	Length	Width	Group	Item Priority	Group Priority
1	258	131	2	1	1
2	312	137	1	1	2
3	210	100	1	2	2
4	202	96	1	3	2
5	341	144	3	1	3
6	136	86	3	2	3

Figure 1 contrasts two layouts for this instance, each solved to optimality under different objectives using the mathematical framework detailed later in this paper. The left panel shows the optimal layout generated when the objective is solely to minimize the total weighted rectilinear distance from each item's centroid to the access point (white circle). The weights guiding this proximity objective are derived from a combination of the 'Group Priority' and 'Item Priority' values listed in Table 1, reflecting each item's overall importance for being near the access point (the specific calculation method is described in Section 5.1.1). While the resulting arrangement might not perfectly match intuitive visual placement based purely on rank (e.g., item 1 is not the absolute closest), this layout represents the mathematical optimum found by the solver when only these access point proximity weights are considered.

The right panel shows the optimal layout produced by our full prioritization approach. This layout minimizes the same weighted distance to the access point as the left panel, but also incorporates an incentive for items belonging to the same group to be placed closer together.

This 'group cohesion' aspect is driven by relative priorities among items within the same group, based on the 'Item Priority' values from Table 1 (as detailed in Section 5.1.1). The resulting layout demonstrates how our approach encourages items within the same group (indicated by shading: white=highest priority group, gray=medium, black=lowest) to cluster together, while still positioning high-priority items advantageously near the center-left access point. This simple example highlights how incorporating group cohesion via our prioritization matrix leads to operationally superior layouts while still considering access priorities.

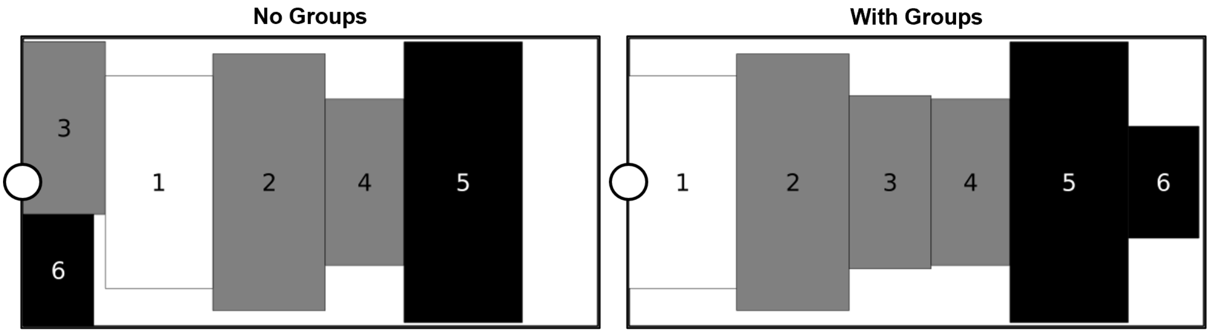

Figure 1: Comparison of layouts for the 6-item example. Left: Layout optimizing only for access point proximity (No Groups). Right: Layout optimizing for both access point proximity and group cohesion (With Groups). Items are shaded by group (white, gray, black) and labeled by overall priority rank (1 = highest, 6 = lowest). The access point (center-left edge) is indicated by the white circle.

Prioritization-based packing has wide-ranging applications in commercial settings such as warehouse operations and cargo stowage. In these contexts, a single shipment or customer order may involve multiple items that form a distinct group, which needs to be kept cohesive to simplify downstream handling. Within each group, certain items might also require a specific sequence for picking or unloading, such as fragile goods that must be accessible first or last. Finally, across multiple groups of items, some groups can be more time-critical than others (e.g., in a vehicle routing solution for delivery truck shipments), creating a tiered prioritization among different groups. Modern e-commerce fulfillment centers exemplify this multi-level prioritization. Optimizing item placement within storage bins or shipping containers is critical to reducing retrieval times and improving operational efficiency. In its warehouse operations, UPS conducts detailed inventory slotting analysis to determine the best placement of items for selection, replenishment, and workload balance, while constantly evaluating facility layout designs (UPS, 2023). Both UPS and DHL increasingly deploy automation and robotics technologies in warehouses to increase process efficiency (DHL, 2023), yet must also juggle the trade-off between placing items near loading points for faster dispatch and preserving group cohesion. Similar challenges arise in freight transportation, where cargo must be positioned strategically to minimize handling time and the number of trips necessary to move goods. FedEx uses truck-loading robots to help determine ideal package layouts to make the most of available space and subsequently load the packages, saving time and effort (Dexterity AI, 2023). The transportation industry increasingly employs automated load-planning systems that optimize for weight distribution and space efficiency, yet many of these systems lack mechanisms to account for adjacency-based priorities. In warehouse operations, there has been little research studying the impact of human factors in order-picking efficiency (Grosse et al., 2017). Analyzing optimal storage positions of items could help derive guidelines that balance ergonomics and performance goals. Taken together, these examples underscore how packing decisions often require careful sequencing across multiple groups, proper ordering of items within each group, and close proximity among group members—three interconnected levels of priority that go beyond simple space minimization.

A primary motivation for this study arises in military logistics, where many of the same multi-level considerations apply. Combat loading in a contested environment demands rapid and prioritized off-loading, but also close grouping of functionally related assets. As outlined in Joint Publication 3-02 (Joint Chiefs of Staff, 2019) and historical embarkation planning doctrine, the arrangement of cargo on transport platforms must align with the operational priority of deployment on the ground. For example, artillery assets may need to be off-loaded first while remaining in close proximity to supporting vehicles. Existing load planning tools, such as the Integrated Computerized Deployment System (ICODES) Single Load Planner, ensure feasibility in load planning by performing structural checks, such as weight distribution and stability constraints (Goodman and Pohl, 2003). However, these systems typically require extensive manual intervention to satisfy operational priorities, such as enforcing a specific off-loading sequence or maintaining adjacency between mission-critical assets. Without a formal mechanism to incorporate these spatial priorities, planners must resort to trial-and-error efforts that often fail to capture all desired operational nuances (Kirschenman et al., 2024). In particular, groups of functionally related equipment must remain cohesive to rapidly organize upon off-loading, the equipment may have a desired group-internal sequence (e.g., combat vehicles before support equipment), and certain groups may need to off-load before others (e.g., to establish a defensive perimeter at a contested beach or port). Balancing these layered requirements—with minimal space usage and quick access to off-loading or access points—demands an approach that integrates all three levels of priority. Historical military operations underscore the criticality of these principles. For instance, an after-action report regarding embarkation schedules for the Inchon Landing campaign highlighted that in challenging over-the-beach operations with limited off-loading facilities and restricting conditions, "debarkation priorities might well be considered more extensively" and stressed that "it is paramount that the combat elements and their combat material go over the beaches together" (X Corps Headquarters, 1951). This directly mirrors the need to manage both item/group off-loading priorities and maintain group cohesion, which are central tenets of our approach.

In this paper, we concentrate on a single-bin environment rather than addressing which bin to choose or whether items fit in multiple bins. We assume that an upstream process has already determined a suitable bin; if items do not fit, the problem is simply infeasible. Our primary objective, therefore, is to model and optimize how items are arranged inside the chosen bin under multiple spatial priorities. While our approach naturally extends to multi-bin or multi-vessel selection in future work, here we highlight the fundamental challenge of prioritization-aware packing within a single bin.

The core contribution of this work is a reformulation of the two-dimensional bin packing problem that integrates a prioritization-weighted distance objective, enabling users to specify proximity relationships that influence how items are positioned within the bin while maintaining standard packing constraints. We demonstrate the flexibility of this framework by applying it to scenarios with multiple levels of prioritization.

The remainder of this paper is structured as follows. Section 2 reviews the relevant literature, covering classical bin packing formulations, facility layout models, and prioritization-based packing approaches. Section 3 presents the mathematical formulation of the proposed model, adapting prior three-dimensional bin packing research to our two-dimensional prioritization-aware setting. Section 4 discusses solution approaches and lower bounding techniques, while Section 5 presents a computational performance evaluation of these approaches. Finally, Section 6 offers managerial insights, outlines potential extensions, and discusses how the proposed framework can be adapted for broader applications in transportation and logistics.

By integrating prioritization considerations into the bin packing problem objective, this study provides a structured approach to optimizing item placement in scenarios where spatial feasibility and operational priority must be jointly considered.

2. Literature Review

This section summarizes the key bodies of literature relevant to our work. We begin with classical bin packing in Section 2.1, then examine the facility layout literature in Section 2.2, and finally review studies that incorporate prioritization or weighted objectives into packing problems in Section 2.3.

2.1. Bin Packing Literature

Bin packing problems have been widely studied in combinatorial optimization, with applications in logistics, manufacturing, and storage optimization. The fundamental problem consists of packing a set of items into one or more bins while ensuring non-overlapping placement and minimizing either the number of bins used or the amount of unused space in an input minimization context (Wäscher et al., 2007). Even in its simplest one-dimensional form, bin packing is NP-hard in the strong sense (Garey and Johnson, 1979). The complexity increases significantly in higher-dimensional cases where width, height, orientation, and other spatial constraints must be considered (Bortfeldt and Wäscher, 2013).

In the two-dimensional variant (2D-BPP), items must be arranged within rectangular bin boundaries while preventing overlaps, optionally allowing a restricted set of rotations (Lodi et al., 2002). Classical approaches to the 2D-BPP emphasize optimizing space utilization, with many studies focusing on heuristics and exact algorithms for minimizing the number of bins required (Christensen et al., 2017; Lodi et al., 2002; Iori et al., 2021). The geometric nature of the problem has led to a variety of solution methods, including mixed-integer linear programming (MILP) formulations (Castro and Oliveira, 2011; Delorme et al., 2016), heuristic placement strategies (Chazelle, 1983; Coffman et al., 1999), and metaheuristic techniques such as tabu search, Greedy Randomized Adaptive Search Procedure (GRASP), or genetic algorithms (Hopper and Turton, 2001; Alvarez-Valdés et al., 2013; Lodi et al., 1999).

While much of the bin packing literature focuses on minimizing bin count or maximizing space usage, certain variants introduce additional considerations, such as weight balancing, load stability, or sequence-based objectives (Li and Zhang, 2018). In particular, residual bin packing problems, which involve strongly heterogeneous items and bins, have received comparatively less attention (Wäscher et al., 2007). The problem considered in this work can be classified as a single-bin residual bin packing problem, where all items must be packed into a single, predefined bin, while the objective extends beyond space utilization to include prioritization-weighted distances between items and an access point on the bin's edge.

To model the necessary geometric constraints, we follow prior three-dimensional bin packing formulations by Paquay et al. (2016) and Fontaine and Minner (2023), adapting them to a two-dimensional setting. Fontaine and Minner (2023) brought further computational enhancements, such as distance normalization and specialized symmetry-breaking constraints, but these improvements are tailored to standard bin packing objectives and do not readily apply to our distance-based prioritization. We retain the core non-overlapping and orientation constraints from these studies while introducing a prioritization-weighted distance objective to handle various types of spatial priorities—such as pulling certain items closer to a bin access point, or encouraging adjacency among high-priority pairs of items—even though the underlying bin packing constraints remain similar to classical 2D-BPP approaches.

2.2. Facility Layout Planning Literature

Facility layout planning (FLP) focuses on the spatial arrangement of departments, workstations, or operational units within a bounded facility to optimize metrics such as material handling costs or adjacency relationships. Traditionally, FLP has been formulated as a combinatorial optimization problem where the objective is to minimize total weighted rectilinear distance while ensuring non-overlapping orthogonal arrangement (Meller et al., 1998). Classical formulations broadly categorize FLP into discrete or continuous models, distinguished by whether department placements are confined to predefined grid positions or allowed to move freely within the facility space (Drira et al., 2007).

One of the earliest mathematical formulations of FLP is the Quadratic Assignment Problem (QAP) (Koopmans and Beckmann, 1957), which assigns departments to fixed grid locations. However, QAP-based methods are often computationally prohibitive for large problems, leading to various heuristic and metaheuristic approaches including simulated annealing (Burkard and Rendl, 1984) and genetic algorithms (Tate and Smith, 1995). More flexible representations, such as the Unequal-Area Facility Layout Problem (UA-FLP), allow departments to have non-uniform sizes and aspect ratios (Bozer and Meller, 1997).

These UA-FLP formulations share geometric constraints with the 2D-BPP, particularly where departments or items must fit within a finite space. However, UA-FLP typically focuses on minimizing total distance-based cost (e.g., flow costs between departments). Most facility layout models do not explicitly capture multi-tiered prioritization: for instance, certain items might need strict adjacency or sequential access—scenarios found in contested military settings. Studies on multi-floor layout problems address both horizontal and vertical adjacency (Ahmadi et al., 2017), but these typically focus on assigning departments to floors rather than embedding more granular priority relationships among individual departments. Similarly, the literature on warehouse design sometimes accounts for pick frequency or accessibility (Gu et al., 2007), yet rarely incorporates a flexible adjacency matrix that assigns varying weights to pairwise item and item-to-access-point proximities to assist with actual item placement in the storage location assignment problem. This gap leaves many real-world applications outside the scope of traditional facility layout approaches.

Accordingly, bridging 2D-BPP constraints with distance-centric FLP objectives offers a way to handle multi-level prioritization. In our work, we place a structured prioritization-weight matrix into a bin packing formulation, thus capturing adjacency-based costs akin to FLP while retaining classical constraints on item placement and orientation.

2.3. Spatial Optimization in Bin Packing and Related Problems

In many real-world applications, the placement of items within a bin is influenced by priorities beyond simple space utilization. In particular, item accessibility, load balancing, sequencing, and adjacency preferences introduce additional complexity into bin packing formulations. Unlike studies that consider prioritizing which bin an item is assigned to, the present work focuses on prioritizing the internal arrangement of items within the bin. These priority-driven ideas have been explored in multiple domains, including container loading, multi-drop delivery, and facility layout problems.

Container loading problems (CLP) provide one of the most direct extensions of bin packing in practical logistics, particularly in settings where item placement influences the efficiency of sequential unloading. In multi-drop container loading, for instance, cargo must be arranged to allow staged unloading at different destinations, ensuring that items for the first drop-off points are more accessible than those required later (Bischoff and Ratcliff, 1995). Several studies explicitly incorporate prioritization: Altarazi (2013) proposes a two-step heuristic for truck loading with priority weights based on demand, while do Nascimento et al. (2021) decompose the CLP into subproblems to deal with feasibility, ensuring a strict prioritization hierarchy for packing considerations similar to Vancroonenburg et al. (2014).

A broader review of container loading constraints highlights that prioritization is rarely explicitly modeled as an optimization objective. Bischoff and Ratcliff (1995) mention that relative priorities could be handled by adjustments of the coefficients in the objective function, but this only relates to the inclusion of an item in a solution — not relative distances between items or to a bin's access point. A survey by Bortfeldt and Wäscher (2013) found that less than 2% of container loading papers at the time explicitly considered loading priorities in the design of algorithms for container loading. Filella et al. (2023) take an innovative approach by modeling unloading constraints as soft constraints, introducing penalties in the objective function and thus more flexibly optimizing both for space utilization and desired loading configurations. However, pairwise item interactions are not considered.

Another related stream of research emphasizes load balancing and stability in bin packing. Trivella and Pisinger (2016) introduce the load-balanced multi-dimensional bin packing problem, formulating the objective function to keep the center of mass of the loaded bins as close as possible to a desired barycenter location while also minimizing the number of total bins used. Erbayrak et al. (2021) extend this idea into a multi-objective approach that also promotes "family unity," meaning items from the same product family end up in the same bin. However, such family unity does not ensure close adjacency within the bin, nor does it account for sequential ordering or varying strengths of cohesion within a family of items. By contrast, the method we propose encourages more nuanced relationships: certain groups of items might need to cluster tightly, others might require moderate proximity, and some items might need to lie near a bin edge for rapid access rather than purely for space efficiency.

Overall, as far as we know, existing bin packing or container loading models do not incorporate a flexible distance-based prioritization matrix that simultaneously captures item-to-item and item-to-bin relationships. By integrating these priorities directly into the objective, our work unifies key ideas from both bin packing and facility layout planning. This provides a framework for important use cases such as warehouse design, multi-drop delivery scenarios, and military combat loading, where the relative positioning of items critically impacts retrieval efficiency and overall operational effectiveness.

3. The 2-D Bin Packing Problem with Prioritization

We consider a single rectangular bin of fixed length L and width W, into which a set of n rectangular items $i \in I = \{1, 2, \ldots, n\}$ must be placed. Each item i has length p_i and width q_i, potentially oriented in either dimension (rotated 90°). Unlike classical two-dimensional bin packing, which aims solely to minimize unused space or the number of bins, here we incorporate a prioritization-based objective that draws certain items together or pulls them toward a designated bin access point on the bin's boundary. We formulate this as a MILP, using efficient formulations introduced by Paquay et al. (2016) and extended by Fontaine and Minner (2023), but simplifying to a single-bin, 2-D setting, with a customized distance-based objective. We highlight the Fontaine and Minner (2023) formulation's parameter and variable naming, as the naming differs slightly between the two papers.

The 2D-BPP with prioritization is subject to fundamental constraints: (1) all items must be placed entirely within the boundaries of the assigned bins; (2) no two items assigned to the same bin may overlap in any dimension; and (3) items must be placed orthogonally to each other, such that each dimension of an item is parallel to one of the bin dimensions. While additional constraints can be incorporated (e.g., item rotation restriction or load balancing), they are excluded from this study to focus on a generalized formulation of the problem.

We define a prioritization matrix π_{ik} for each item $i, k \in I$ where $i \leq k$. We encode a prioritization weight structure within π_{ik} that allows for different packing configurations. We compute π_{ik} ahead of time and include it as a parameter to our model. For π_{ik} where $i < k$, π_{ik} encodes the priority weight for items i and k to be close to each other. For π_{ii}, we encode the priority weight for placing item i close to the bin access point. In Section 4.2.1 we introduce a formal assumption regarding the construction of these weights. Let (x^o, y^o) denote the coordinates of the bin access point, which is treated as a known parameter of the model.

Each item $i \in I$ has a continuous x_i (y_i) variable designating the coordinate position of the left (bottom) side of the item; a continuous x_i^r (y_i^r) variable designates the coordinate position of the right (upper) side of the item. The binary variables a_{ik}^x (a_{ik}^y) are equal to 1 if item i is placed entirely to the left (bottom) of item k in the x- (y-) dimension, and 0 otherwise. $t_{i,c,d}$ defines the orientation of item i with respect to the bin. Here, $C = \{1, 2\}$ corresponds to the dimensions (x, y) of the bin, and $D = \{1, 2\}$ corresponds to the dimensions (x, y) of the item. We introduce auxiliary decision variables ρ_{ik}^x and ρ_{ik}^y to represent rectilinear distances, whether between centroids of each item i and k where $i < k$, or between each item i and the associated bin access point (x^o, y^o) when $i = k$. We use these rectilinear distances in the objective function to spatially optimize the placement of items within the bin. The mathematical model is then defined as follows:

$$
\begin{aligned}
\min \quad & \sum_{i \in I} \sum_{k \in I,\, i \leq k} \pi_{ik} \big(\rho^x_{ik} + \rho^y_{ik} \big) && && (1)\\
\text{s.t.} \quad & x^r_i - x_i = t_{i,1,1} p_i + t_{i,1,2} q_i && \forall i \in I && (2)\\
& y^r_i - y_i = t_{i,2,1} p_i + t_{i,2,2} q_i && \forall i \in I && (3)\\
& a^x_{ik} + a^x_{ki} + a^y_{ik} + a^y_{ki} \geq 1 && \forall i, k \in I, i < k && (4)\\
& \sum_{c \in C} t_{i,c,d} = 1 && \forall i \in I, d \in D && (5)\\
& \sum_{d \in D} t_{i,c,d} = 1 && \forall i \in I, c \in C && (6)\\
& x^r_k \leq x_i + (1 - a^x_{ik}) L && \forall i, k \in I && (7)\\
& x_i + 1 \leq x^r_k + a^x_{ik} L && \forall i, k \in I && (8)\\
& y^r_k \leq y_i + (1 - a^y_{ik}) W && \forall i, k \in I && (9)\\
& y_i + 1 \leq y^r_k + a^y_{ik} W && \forall i, k \in I && (10)\\
& \frac{x_i + x^r_i}{2} - \frac{x_k + x^r_k}{2} \leq \rho^x_{ik} && \forall i, k \in I,\ i < k && (11)\\
& \frac{x_k + x^r_k}{2} - \frac{x_i + x^r_i}{2} \leq \rho^x_{ik} && \forall i, k \in I,\ i < k && (12)\\
& \frac{y_i + y^r_i}{2} - \frac{y_k + y^r_k}{2} \leq \rho^y_{ik} && \forall i, k \in I,\ i < k && (13)\\
& \frac{y_k + y^r_k}{2} - \frac{y_i + y^r_i}{2} \leq \rho^y_{ik} && \forall i, k \in I,\ i < k && (14)\\
& \frac{x_i + x^r_i}{2} - x^o \leq \rho^x_{ik} && \forall i, k \in I,\ i = k && (15)\\
& x^o - \frac{x_i + x^r_i}{2} \leq \rho^x_{ik} && \forall i, k \in I,\ i = k && (16)\\
& \frac{y_i + y^r_i}{2} - y^o \leq \rho^y_{ik} && \forall i, k \in I,\ i = k && (17)\\
& y^o - \frac{y_i + y^r_i}{2} \leq \rho^y_{ik} && \forall i, k \in I,\ i = k && (18)\\
& t_{i,c,d}, a^x_{ik}, a^y_{ik} \in \{0, 1\} && \forall i, k \in I, c \in C, d \in D && (19)\\
& 0 \leq x_i \leq x^r_i \leq L && \forall i \in I && (20)\\
& 0 \leq y_i \leq y^r_i \leq W && \forall i \in I && (21)\\
& \rho^x_{ik},\ \rho^y_{ik} \geq 0 && \forall i, k \in I,\ i \leq k && (22)
\end{aligned}
$$

The objective (1) sums the weighted rectilinear distances among items or between each item and the bin's access point, where weights come from π_{ik}. This structure pulls specified items closer together when their pairwise importance warrants adjacency, while simultaneously drawing each item toward the bin access point according to its individual priority level. For each item, Constraints 2 and 3 collectively ensure each item's coordinates reflects the actual item dimensions given the chosen item orientation. Constraint 4 ensures no overlapping of items i and k. Constraints 5 and 6 ensure that each item's dimension is aligned to one of the bin's dimensions, effectively allowing for $90°$ rotation. Constraints 7 and 8 geometrically ensure if item i is to the right (left) of item k, the right (left) side of item k must be less (greater) than the left (right) side of item i. Constraints 9 and 10 do the same by ensuring if item i is to the top (bottom) of item k, the upper (bottom) side of item k must be less (greater) than the bottom (upper) side of item i. Constraints 11–14 define ρ_{ik}^x and ρ_{ik}^y as absolute rectilinear distances between the centroids of items i and k, and (15)–(18) likewise measure the distance between the centroid of each item i and (x^o, y^o) for the diagonal entries π_{ii}. Constraints 19–22 establish decision variable domains.

Although many bin packing and facility layout formulations incorporate explicit symmetry-breaking constraints (e.g., forcing an item to one half of the bin), we forgo such constraints in this work. In most cases, the bin's access point already provides sufficient asymmetry unless it lies exactly on a midline, and even then, small-ε offsets for a single item can locally break symmetry without unduly restricting the feasible region. Adding more complex logic-based constraints to eliminate all mirrored solutions would significantly expand the branch-and-bound tree, undercutting potential computational gains. Hence, we accept the small risk of mirrored layouts in exchange for a simpler model that preserves advantageous placements along the midline when they are beneficial.

Even though our formulation reduces the three-dimensional model from Fontaine and Minner (2023) into two dimensions, the introduction of the prioritization structure in our objective function significantly complicates the optimization. Specifically, the interplay between group-level priorities, item-level sequencing preferences, and spatial proximity requirements leads to complex trade-offs. These trade-offs prevent the use of traditional relaxations and drastically increase computational difficulty, making even moderate-sized problem instances challenging or impossible to solve to optimality within reasonable time limits.

4. Solution Techniques and Lower Bounds

In this section, we present the core methods we use to solve and evaluate the two-dimensional bin packing problem with prioritization. Section 4.1 outlines two primary solution approaches: a direct single-MILP model and a sliding-window decomposition strategy that exploits group-based and item-based rankings. Then, in Section 4.2, we discuss two lower-bound techniques designed to assess solution quality in a more rigorous manner. We highlight both the algorithmic and theoretical aspects of our framework before presenting numerical experiments in Section 5.

4.1. Solution Techniques

Having defined how we incorporate different priority levels, we now present and compare two primary methods for solving the prioritized 2-D BPP introduced in Section 3. We focus on a single-bin environment where items can belong to different groups, and each group or item has an associated rank indicating its relative importance. These priorities guide how we structure subproblems in the following approaches.

In preliminary testing, we also explored a "groupwise decomposition" that solved entire groups in descending order of priority. While intuitive and conceptually simpler, that strategy did not allow items from different groups to be rearranged together, leading to consistently worse solutions and longer run times compared to the sliding-window approach. Thus, we omit further details here.

4.1.1. Single MILP Approach

A straightforward approach to finding a solution is to solve the full 2-D BPP with prioritization as a single, unified MILP. Conceptually, one feeds the entire set of items into the model outlined in Section 3 and utilizes a general-purpose solver (e.g., Gurobi). This method is referred to here as the Single MILP Approach. We construct the prioritization-based objective for all items $i, k \in I$, activate all item-placement and non-overlap constraints, and then run the solver until it either proves optimality or reaches a prescribed time limit.

A main advantage of the single MILP approach is that, without a time limit, it can prove a globally optimal solution. Even with time limits, it yields a bounded optimality gap if the solver can establish both primal and dual bounds. However, computation times grow rapidly as the number of items increases. In practice, for instances beyond 7 items, the solver may require hours to confirm optimality, even if it finds a feasible solution quickly. Hence, although this direct method is conceptually straightforward, it often proves intractable for larger problems in real-world planning horizons.

4.1.2. Sliding-Window Decomposition Approach

Given the computational challenges of the single MILP approach, decomposition methods offer a way to break the problem into smaller, more manageable subproblems. One natural attempt is to consider groups individually based on priority. However, as noted earlier, fixing entire groups sequentially can lead to suboptimal global arrangements. Therefore, we focus on a sliding-window decomposition approach.

This decomposition approach focuses on a fixed number of items at a time, rather than the entire set of items. By concentrating on only a handful of items in each subproblem, it avoids ever having to solve a large single MILP.

We begin by ordering all items in descending priority, drawing on both group affiliation and item-level ranks to produce a single global list. We choose a window size w and solve for the top w items under the usual constraints and objective. Once this subproblem terminates, we fix the single highest-priority item among those w items and keep the other $(w - 1)$ items "free." We then slide the window forward by introducing the next item in the global priority list (i.e., the $(w + 1)$-th item), maintaining the subproblem size at w. Because we fix one old item and add one new item each iteration, only w items remain truly variable, making it less computationally taxing than a full-instance solve.

A key parameter is the sliding window size, w. Our experiments vary w from 1 to 9. Naturally, $w = 1$ is an extreme that places items one by one (akin to a simple greedy strategy), whereas a larger w captures more local packing trade-offs at the cost of potentially longer solve times per iteration. Because these smaller subproblems can each be run with a time limit, the method can progress quickly through the globally ordered priority list, freeing and fixing items iteratively.

In this decomposition, each subproblem is kept small, typically leading to shorter runtimes. By sliding through a single global priority list, one can occasionally re-optimize items from different groups in the same window, offering cross-group flexibility. However, each iteration fixes only the top-priority item and locks its coordinates, potentially "freezing" suboptimal placements early if the window is too small. Performance thus depends heavily on choosing a window size that balances computational effort and solution quality.

4.2. Lower Bound Methods

We now discuss two techniques to establish lower bounds for the prioritized 2-D BPP. In the first, a grid-based geometric construction leverages Euclidean distances to guarantee a valid underestimation of rectilinear costs. In the second, a statistical approach applies extreme value theory to infer a lower bound from the distribution of observed feasible minima. Both methods provide benchmarks for assessing how close our solution approaches come to optimality, especially when exact proofs of optimality are computationally infeasible.

4.2.1. Grid-Based Lower Bound Construction

We first develop a specialized grid-based lower bound (GB-LB) procedure to evaluate solution quality, especially for large instances where proving optimality is intractable. Standard bin packing bounds often fail here due to the distance-based objective and prioritization structure. Our GB-LB leverages the geometric properties of the problem, using Euclidean distances—which are inherently less than or equal to the rectilinear distances minimized in our objective function ($\|u-v\|_2 \leq \|u-v\|_1$)—to ensure a valid underestimation of the true optimal cost (OPT). We construct lower bounds ρ_{io}^{LB} for item-to-access-point distances and ρ_{ik}^{LB} for item-to-item distances, such that the resulting weighted sum $\text{LB} = \sum \pi_{io}\rho_{io}^{LB} + \sum \pi_{ik}\rho_{ik}^{LB}$ satisfies $\text{LB} \leq \text{OPT}$.

We consider n items ($i \in I = \{1, \ldots, n\}$) with integer dimensions (p_i, q_i) within a bin discretized into 1×1 cells. Each item i occupies $p_i \times q_i$ distinct cells. A single access point o lies on the bin boundary. The process for determining ρ_{io}^{LB} is illustrated in Figure 2. The left panel of the figure shows the original dimensions of the first four example items from Table 1, while the right panel depicts the conceptual packing process. As higher-priority items are notionally packed by placing their discretized 1×1 unit cells as close as possible to the access point o, they form radially expanding exclusion zones.

In order to derive and prove Theorem 1 (GB-LB), we rely on the following assumption regarding the prioritization structure:

Assumption 1. *Items possess distinct access-point priorities, allowing unique relabeling such that* $\pi_{1o} > \pi_{2o} > \cdots > \pi_{no}$.

This assumption implies a clear, strict ordering of items based on the criticality of their proximity to the access point. This strict ordering enables the unambiguous greedy placement procedure used in the proof of Theorem 1 below, which is essential for the validity of the lower bound construction. As discussed in Section 5.1.1, this type of structure is plausible in scenarios where individual item access is paramount, effectively creating a hierarchy where item and group

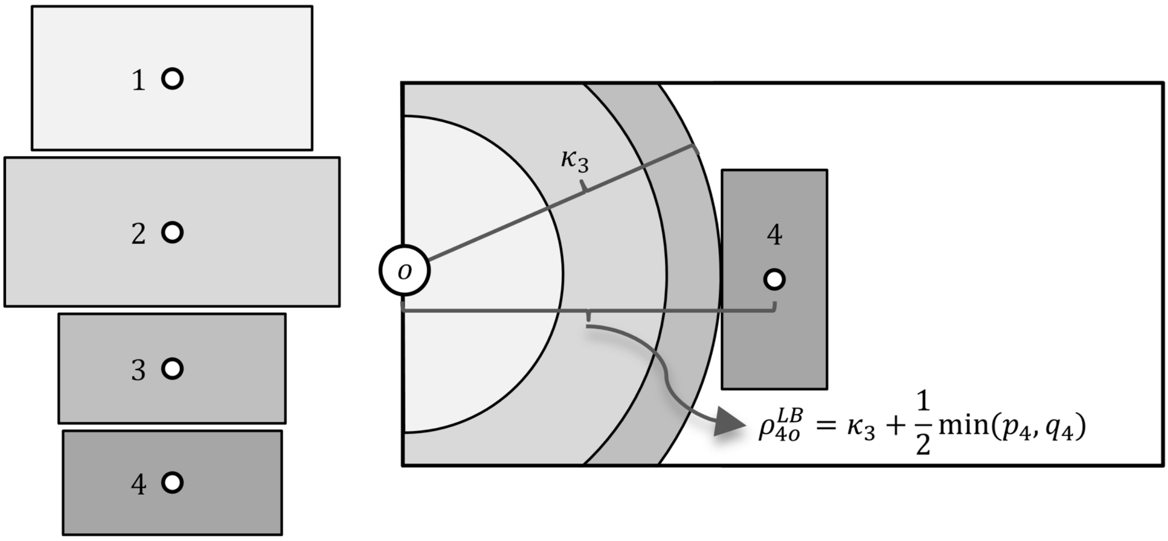

Figure 2: Grid-Based Lower Bound (GB-LB) item-to-access-point distance calculation. Left: Scaled original shapes and centroids of example items 1-4 (from Table 1). Right: Conceptual radial packing of items 1-3 (lighter grays) from access point o within a discretized bin, forming exclusion radius κ_3. Item 4 (darker gray) is shown pre-discretization to visualize how its lower bound distance $\rho_{4,o}^{LB}$ is determined using κ_3 and its minimum dimension.

priorities for access point proximity dominate group cohesion incentives for the purpose of this bound calculation. We emphasize that Assumption 1 is required only for deriving this specific lower bound.

Theorem 1 (Grid-Based Lower Bound). *Consider a bin discretized into 1×1 cells, a bin access point o on the bin boundary, and n items satisfying Assumption 1 (i.e., sorted such that $\pi_{1o} > \pi_{2o} > \cdots > \pi_{no}$). Let ρ_{io}^{LB} and ρ_{ik}^{LB} be distances constructed as defined in Equations (23)-(25). Then the lower bound* LB*, calculated as*

$$\mathrm{LB} \;=\; \sum_{i=1}^{n} \pi_{io}\, \rho_{io}^{LB} \;+\; \sum_{i<k} \pi_{ik}\, \rho_{ik}^{LB},$$

satisfies $\mathrm{LB} \leq \mathrm{OPT}$*, where* OPT *is the minimal objective value using rectilinear distances for any feasible packing.*

Proof. The proof proceeds by constructing the distances ρ_{io}^{LB} and ρ_{ik}^{LB} and then arguing their validity as lower bounds on the corresponding actual rectilinear distances in any feasible solution.

First, discretize the bin and items: Partition the bin into 1×1 cells. Represent each item i as its $p_i \times q_i$ constituent unit squares, which must occupy distinct cells.

Second, determine item-to-access-point bounds ρ_{io}^{LB} via iterative placement: Process items sequentially according to the priority order guaranteed by Assumption 1 ($i = 1, \ldots, n$). For item i, determine the minimum possible Euclidean distance its centroid can be from o, given the space occupied by higher-priority items $1, \ldots, i-1$. Let κ_{i-1} be the radius of the largest circle centered at o such that all cells with centroids strictly inside it are notionally occupied by items $1, \ldots, i-1$ (with $\kappa_0 = 0$). If $i > 1$, find the closest unoccupied cell centroid to o (at distance ρ^*) and define the exclusion radius κ_{i-1} as:

$$\kappa_{i-1} = \rho^* - \tfrac{\sqrt{2}}{2}. \tag{23}$$

Item i's centroid must be at least half its minimum dimension away from this radius. Define the lower bound distance as:

$$\rho_{io}^{LB} \; = \; \kappa_{i-1} \; + \; \tfrac{1}{2}\,\min\{p_i, q_i\}. \tag{24}$$

Conceptually place item i's cells in the closest available cells outside radius κ_{i-1} before proceeding to item $i+1$.

Third, determine item-to-item bounds ρ_{ik}^{LB}: For any pair (i,k) with $i<k$, the minimum possible Euclidean distance between their centroids, due to non-overlap constraints and their physical dimensions, occurs when they are adjacent along their shortest sides. Define this conservative lower bound as:

$$\rho_{ik}^{LB} \; = \; \tfrac{1}{2}\,\min\{p_i, q_i\} \; + \; \tfrac{1}{2}\,\min\{p_k, q_k\}. \tag{25}$$

Fourth, establish validity: Let $(\rho_{io}^x + \rho_{io}^y)$ and $(\rho_{ik}^x + \rho_{ik}^y)$ be the actual rectilinear distances in an optimal packing. Let ρ_{io}^{Euc} and ρ_{ik}^{Euc} be the corresponding actual Euclidean distances in that same packing. By the iterative construction in the second step, the exclusion radius κ_{i-1} (Equation (23)) forces item i to be at least ρ_{io}^{LB} (Equation (24)) away from o in Euclidean terms. Thus, $\rho_{io}^{LB} \le \rho_{io}^{\text{Euc}}$. By the non-overlap requirement and the definition in the third step (Equation (25)), the minimum separation forces $\rho_{ik}^{LB} \le \rho_{ik}^{\text{Euc}}$. Since Euclidean distance never exceeds rectilinear distance, $\rho_{io}^{\text{Euc}} \le (\rho_{io}^x + \rho_{io}^y)$ and $\rho_{ik}^{\text{Euc}} \le (\rho_{ik}^x + \rho_{ik}^y)$. Combining these yields $\rho_{io}^{LB} \le (\rho_{io}^x + \rho_{io}^y)$ and $\rho_{ik}^{LB} \le (\rho_{ik}^x + \rho_{ik}^y)$. Given the non-negative weights $\pi_{io}, \pi_{ik} \ge 0$, we can sum the weighted lower bounds:

$$\sum_{i=1}^{n} \pi_{io}\,\rho_{io}^{LB} + \sum_{i<k} \pi_{ik}\,\rho_{ik}^{LB} \;\le\; \sum_{i=1}^{n} \pi_{io}\,(\rho_{io}^x + \rho_{io}^y) + \sum_{i<k} \pi_{ik}\,(\rho_{ik}^x + \rho_{ik}^y)$$

The left side is the definition of LB and the right side is OPT. Thus, LB $\le$ OPT, proving the theorem. □

This GB-LB is particularly effective when item-to-access-point priorities (π_{io}) dominate. Its weakness lies in the ρ_{ik}^{LB} calculation, which only considers pairwise minimal dimensions and can be loose if group cohesion weights (π_{ik}) are significant and items are not tightly packed. However, its computational simplicity provides a valuable benchmark, often improving upon Gurobi's bound as problem size increases.

4.2.2. Statistical Bound using Extreme Value Theory

Having discussed a deterministic bound, we also use a complementary statistical lower bound (Wilson et al., 2004) to further evaluate solution quality. In essence, Wilson's method leverages extreme value theory, drawing upon the fundamental result by Fisher and Tippett (1928) regarding the asymptotic distribution of extreme values, to fit a three-parameter Weibull distribution to observed minimum objective values, thereby constructing confidence intervals for the global optimum z. The three-parameter Weibull cumulative distribution function (CDF) is given by $F(x; a, b, c) = 1 - \exp\left(-\left(\frac{x-a}{b}\right)^c\right)$ for $x \geq a$, where a is the location (threshold) parameter representing the minimum possible value, $b > 0$ is the scale parameter, and $c > 0$ is the shape parameter.

Our implementation proceeds as follows. First, for each large problem instance we create n "slightly perturbed" variants by altering only the per-group priority hierarchies, keeping all other aspects of the problem constant. Each variant is solved using whichever solution method performed best on that instance, and we record the minimum objective value obtained, denoted x_i for $i = 1, \ldots, n$. Crucially, we then recalculate that solution's objective value using the original problem's prioritization hierarchy, so that all minima are assessed on the same baseline. Let $x_{(k)}$ denote the k-th order statistic such that $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$, hence $x_{(1)} = \min\{x_i\}$. Next, we fit the Weibull curve via least-squares estimation to the n observed minima. We use the analytical parameter estimates from Zanakis (1979) as starting values for the location ($\hat{a}$), scale ($\hat{b}$), and shape ($\hat{c}$) parameters:

$$\hat{a} = \frac{x_{[1]}x_{[2]} - x_{[2]}^2}{x_{[1]} + x_{[n]} - 2x_{[2]}}, \tag{26}$$

$$\hat{b} = x_{[\lceil 0.63n \rceil]} - \hat{a}, \tag{27}$$

$$\hat{c} = \frac{\ln\{\ln(1 - q_1)/\ln(1 - q_2)\}}{\ln\{(x_{[\lceil nq_1 \rceil]} - \hat{a})/(x_{[\lceil nq_2 \rceil]} - \hat{a})\}}, \tag{28}$$

where $q_1 = 0.97366$ and $q_2 = 0.16731$. These initial estimates are then refined using the Nelder and Mead (1965) algorithm to obtain the final fitted parameters $\hat{a}, \hat{b}, \hat{c}$. We then evaluate goodness of fit with both the Kolmogorov–Smirnov (KS) and Anderson–Darling (AD) tests, the latter being critical for tail accuracy. Following Golden and Alt (1979), using the fitted parameters, we construct an approximate $100(1-e^{-n})\%$ confidence interval for the threshold parameter a. This interval takes the form $[\hat{x}^*, x_{(1)}]$, where the lower limit $\hat{x}^*$ is given by $\hat{x}^* = x_{(1)} - \hat{b}$. Since the threshold parameter a represents the theoretical minimum objective value, $\hat{x}^*$ provides a statistical lower bound for the true global optimum z. This process largely mirrors the heuristic-based method in Wilson et al. (2004) but adapts naturally to our mixed-integer optimization setting. This general methodology, leveraging extreme value theory to estimate confidence intervals for the optimal objective value, represents a well-established approach in combinatorial optimization for evaluating solution quality where tight deterministic lower bounds may be unavailable or computationally prohibitive (Derigs, 1985; Altinel et al., 2000; Akyüz et al., 2010).

In Section 5.2.3, we illustrate how this method, in tandem with the GB-LB, offers a more comprehensive view of solution quality.

5. Performance Evaluation of Solution Approaches

This section presents a computational study to demonstrate how the proposed 2-D BPP with prioritization performs under realistic logistics scenarios.

5.1. Data, Instances, and Prioritization Matrix

First, we introduce how we construct the prioritization matrix for item and group level priorities, along with group cohesion. Next, we describe our instance-generation procedures, including group and item sizes, item aspect ratios, and bin dimensions. We then outline our computational settings and experimentation process.

5.1.1. Constructing the Prioritization Matrix

In many operational scenarios, there is a need to place higher-priority items close to a bin access point for rapid retrieval or dissemination, while also grouping related items together. To reflect these goals, we use a prioritization matrix π that encodes both group cohesion and access-point proximity. Minor adjustments to π can accommodate certain loading or stowage contexts with different adjacency or distance requirements.

Each diagonal entry π_{ii} is an inverted global priority level, capturing how strongly item i should be placed near the bin's access point, while each off-diagonal entry π_{ik} (for $i < k$) encodes how strongly items in the same group should be drawn together based on their relative priorities. This single matrix π thus unifies proximity to the access point and group cohesion under one set of objective coefficients and can be recalculated whenever group structures or item-level priorities change. To illustrate this construction, using the data for the example instance shown in Table 1, we present the process applied in our computational study. Note that this specific construction method, by assigning unique global ranks which are then inverted to form the diagonal entries π_{ii}, inherently yields distinct access-point priorities for each item. This satisfies Assumption 1 as required for the validity of the Grid-Based Lower Bound developed in Section 4.2.1. Furthermore, this construction emphasizes access-point proximity within the objective function itself, primarily through the relatively larger magnitude of the diagonal π_{ii} entries compared to the off-diagonal π_{ik} entries. Other settings might motivate slightly different constructions of the π matrix.

Overview of Key Elements. π_{ii} (diagonal entries) represent how strongly item i should be placed near the bin's access point. A higher diagonal value indicates a stronger pull toward that location. π_{ik} for $i < k$ (off-diagonal entries) represent how strongly items i and k should be close to each other. In the baseline structure presented here, $\pi_{ik} > 0$ only if i and k belong to the same group.

Step-by-Step Construction. We first sort all groups in ascending order of group-level priority, then sort the items in each group by ascending item-level priority. In this scheme, an item labeled with a smaller numeric value is considered higher priority. This sorting yields a provisional rank for each item, which we then reverse—that is, we flip the ordered list of raw ranks $(1, 2, \ldots, |I|)$ to $(|I|, |I| - 1, \ldots, 1)$ while keeping each item's index attached to its position—so that the item with priority level 1 (i.e., the highest-priority item overall) ends up with the largest rank value. Concretely, if each item i is assigned a raw rank $\alpha[i]$ such that $\alpha[i] = 1$ corresponds to the highest-priority item and $\alpha[i] = 2$ the next, and so on, we set $\alpha^{\text{rev}}[i] = |I| - \alpha[i] + 1$, making $\alpha^{\text{rev}}[i]$ larger for items of higher priority. The reversed rank $\alpha^{\text{rev}}[i]$ becomes the diagonal entry π_{ii}, thereby assigning stronger pull toward the bin's access point for items with higher overall priority.

Next, we define off-diagonal entries π_{ik} for pairs (i, k) belonging to the same group. Let each item i have an integer item-level priority within its group (again, a smaller number indicates higher priority), and define Δ_{ik} to be the absolute difference between those two item-level priorities. We identify $G_{\max}$ as the number of items in the largest group across the entire problem instance. For any two items i and k in the same group, we then set $\pi_{ik} = \frac{G_{\max} - \Delta_{ik}}{G_{\max}}$ which is always strictly between 0 and 1 for nonzero Δ_{ik}. Two items whose item-level priorities differ only slightly thus receive a larger off-diagonal weight, while items farther apart in priority receive a smaller weight. By using the same denominator $G_{\max}$ for all groups, we standardize the off-diagonal scale throughout the instance. Items in different groups simply receive $\pi_{ik} = 0$, indicating no built-in incentive for them to be close. Note that assigning integer ranks (≥ 1) to π_{ii} and fractional values (< 1) to π_{ik} inherently ensures $\pi_{ii} > \pi_{ik}$ for all i, k where $i < k$. This deliberate scaling maintains a strong emphasis on access-point proximity, promoting dense configurations near the access point, while the smaller π_{ik} weights encourage group cohesion without dominating the primary placement objective.

Illustrative Example Data. The data used to generate the prioritization matrix for the illustrative example is shown in Table 1. This involves six items across three groups with assigned item and group priorities. Using this information, we obtain the prioritization matrix π_{ik} shown in Table 2, where the diagonal entries π_{ii} (in bold) reflect global item-level ranks derived from both item and group priorities, while off-diagonal entries encode intra-group closeness based on the formula above (using $G_{max} = 3$ for this instance).

Table 2: Prioritization matrix π for the example instance (derived from data in Table 1).

Item	1	2	3	4	5	6
1	**6**	0	0	0	0	0
2	0	**5**	2/3	1/3	0	0
3	0	0	**4**	2/3	0	0
4	0	0	0	**3**	0	0
5	0	0	0	0	**2**	2/3
6	0	0	0	0	0	**1**

Higher-priority items (diagonal entries) tend to appear closer to the access point, while items in the same group cluster based on off-diagonal values. In larger instances, balancing these competing priorities often necessitates trade-offs between dense packing and group cohesion; however, including off-diagonal values helps promote group cohesion with little to no detriment to access-point proximity, as we will demonstrate quantitatively in Section 5.2.1.

5.1.2. Instance Generation

To motivate the selection of parameters for our computational study, we base our instance generation on characteristics observed in realistic logistics scenarios, particularly military combat loading onto large vessels, while ensuring applicability to other settings where possible. We generate a diverse set of 90 problem instances to evaluate the proposed 2-D BPP with prioritization. Each instance includes between 1 and 5 groups of items, capturing a variety of possible scenarios. Within each group, items share a common mission or operational function and thus belong together for off-loading or unloading purposes, but they can differ in size or priority. Our instances generally proceed as follows:

Group and Item Counts. First, we randomly select the number of groups to be between 1 and 5. Each group then has a size drawn from one of three categories: (i) small (1–4 items), (ii) medium (5–8 items), or (iii) large (9–12 items). We cap groups at a maximum of 12 items for two reasons: (1) this accommodates a typical platoon-sized echelon of vehicles in military contexts, and (2) from a computational standpoint, 12 is large enough to be challenging yet still meaningful within our solution framework.

This range also aligns with observations from online retail data, where 84% of "orders" (analogous to groups) include four or fewer items (Hübner et al., 2015). For instance, Fontaine and Minner (2023) limit order sizes to 10 items based on a Brazilian e-commerce data set (Olist, 2019) in which the maximum observed order size is 21 items, but over 97% contain just one or two. Consequently, each generated problem instance could contain as few as 1 item or as many as 60 items. These extremes reflect a broad range of scenarios—covering the upper limits typically seen in e-commerce order sizes and matching the capacity of the largest roll-on/roll-off (RO/RO) vessel used by the U.S. Navy (United States Navy, 2021) when we consider up to 5 groups of 12 items each.

Homogeneity Levels for Item Sizes. To capture the varied nature of equipment footprints in real logistics settings, we categorize each group as (i) homogeneous, (ii) weakly heterogeneous, or (iii) strongly heterogeneous in terms of item dimensions. A homogeneous group assigns identical length/width to all items, mirroring situations where equipment is standardized. Weak heterogeneity allows a small number of distinct item sizes within the group, while strong heterogeneity permits a wide range of dimensions, approximating, for instance, a mixture of different vehicle classes or varying package sizes. Despite these differences, all items remain rectangular and may be rotated 90°.

Item Dimensions and Aspect Ratios. All items are rectangular, with length and width drawn so that the aspect ratio lies between 1:1 and 1:3. This range mirrors the footprints of typical military wheeled/tracked vehicles—for example, an M1 Abrams tank has length-to-width close to 2.7:1 (General Dynamics Land Systems, 2025). Some items are small (e.g., a generator set or small package), others large (a tank or large crate).

Bin Dimensions and Extended Length. To ensure adequate space for priority-based placement, we start with a sufficient bin area for feasibly packing assigned items and then inflate the bin length by an additional factor. This approach is conceptually similar to the "open dimension problem" in Wäscher et al. (2007)'s typology, allowing extra space to accommodate loading priorities rather than purely minimizing empty area. In practice, the extended bin length grants more freedom for positioning items around the bin access point (for instance, along one edge or corner), making it easier to isolate the effect of prioritization in the objective.

Bin Access Points. We considered two primary bin access points: a center-left (CL) edge and a bottom-left (BL) corner on the bin's plane. These locations represent common configurations found in logistics, such as centered ramps for beach landings or corner ramps for pier docking in RO/RO operations (United States Navy, 2023, 2021). In our experiments, the two access points produced nearly identical results regarding the effectiveness of the prioritization. Therefore, for brevity, we report only CL-based results in subsequent sections. Furthermore, the CL location can represent centered access points common in other settings, like multi-drop delivery trucks, enhancing the broad applicability of the presented results.

All models were implemented in Julia 1.11.0 and solved by Gurobi 12.0. All experiments were conducted on an Intel Xeon (Skylake) processor, 2.30 GHz, with up to 1 terabyte RAM operating at 2400 MHz, and access to up to 32 cores through the attached virtual machine. For each problem instance, we performed a single MILP solve under a time limit defined as the number of items multiplied by 60 seconds. For the decomposition approach, we applied time limits of 5 seconds, 15 seconds, and 60 seconds per windowed solve. Consequently, for each problem instance, a total of 27 solves is carried out for this decomposition approach for each window size ($\forall w \in \{1, 2, \ldots, 9\}$) and time limit combination.

5.2. Performance of Solution Approaches

This section evaluates our prioritized bin packing framework and solution methods, providing insights on quantitative metrics of runtime, solution quality, and optimality gaps. We first quantify the impact of including group cohesion weights in the objective function. We then analyze runtime behavior and parameter tuning for the decomposition approach, evaluate the effectiveness of the lower bounds, compare the objective values achieved by different solution methods, and explore the resulting trade-off between computation time and solution quality.

5.2.1. Quantitative Impact of Group Cohesion

To complement the visual illustration in Figure 1, we quantitatively analyzed the impact of incorporating group cohesion weights (π_{ik} for $i < k$) into the objective function across our 90 problem instances. Ignoring group cohesion (i.e., setting all $\pi_{ik} = 0$ for $i < k$) raises the total objective value by about 3% on average compared to solutions where cohesion weights are included, largely due to penalizing item separation. More significantly, the item-to-item distance component of the objective inflates by over 16% when cohesion is ignored. This improved clustering occurs with minimal negative impact on access-point proximity; evaluating the proximity-only component ("no-group" objective) shows that including cohesion weights leads to an average improvement of 0.13% and a median difference of 0.00% compared to optimizing for proximity alone. Thus, enforcing group cohesion results in significantly better clustering of related items with effectively no penalty to access-point proximity—reinforcing the value of embedding multi-level priorities into bin packing.

5.2.2. Runtime Analysis and Parameter Tuning

As discussed in Section 4.1, the sliding-window decomposition approach applies time limits to subproblems because the time required to prove optimality grows rapidly with the number of items. Table 3 illustrates this effect for all 7-, 8-, and 9-item groups across our 90 problem instances, which are treated as standalone problems. In these experiments, 22 seven-item groups, 30 eight-item groups, and 23 nine-item groups were extracted—75 standalone instances in total—showing a dramatic increase in solution time as instance size grows.

Figure 3 further shows how the sliding-window approach scales with increasing window size (w) and different per-subproblem time limits (5, 15, or 60 seconds). Subproblems with up to 5

Table 3: Summary statistics for time (in seconds) to prove optimality across all 7–9 item groups treated as standalone instances.

Group Size	Mean	Std	Min	Median	Max
7 Items	9.72	3.88	4.26	8.38	17.32
8 Items	94.56	39.91	30.55	88.51	184.75
9 Items	1279.50	613.31	352.67	1281.83	2853.02

items often solve almost instantly, whereas larger windows (8–9 items) may benefit from longer time limits but then risk hitting diminishing returns. Ultimately, windows of 6–7 items under short time limits emerge as a solid compromise between speed and solution quality. These observations guide our choice of decomposition parameters in the subsequent evaluations.

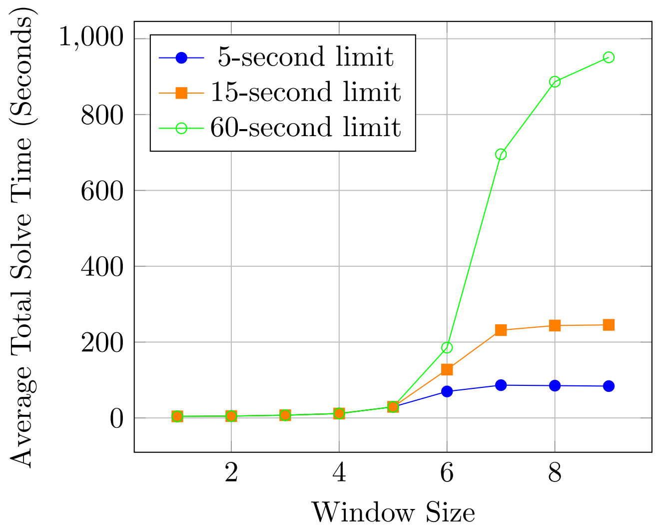

Figure 3: Average Total Time vs. Window Size for Windowed Solve by Time Limit.

5.2.3. Assessing Solution Quality with Lower Bounds

We next investigate how closely our best solutions approach optimality by comparing them against three sets of lower bounds: (1) Gurobi's best lower bound from the single MILP solution approach, (2) the deterministic grid-based lower bound described in Section 4.2.1, and (3) the statistical bound described in Section 4.2.2. We focus here on the largest problem instances since they present the most challenging scenarios and highlight where any bound weaknesses are most pronounced. We note, however, that similar trends hold for smaller problems as well.

Grid-Based Deterministic Bound. Recall from Section 4.2.1 that our specialized grid-based lower bound (GB-LB) explicitly incorporates prioritization through both access-point distances and group-adjacency weights. This approach directly addresses weaknesses inherent to traditional bin packing or facility layout bounds—particularly their inability to account for distance-based prioritization or group-placement constraints. As Table 4 shows, the grid-based lower bound (GB-LB) often provides a much tighter bound, substantially reducing the percentage gap relative to the best-known objective compared to Gurobi's built-in bound (LB), showing that geometrically constructed bounds can have increased utility as problem size grows.

Table 4: Comparison of Best Known Objective and Lower Bounds for 8 Largest Problem Instances. Values for Obj, LB, and GB-LB are scaled by 10^5 (e.g., 11.3 represents 1.13e6). % Gap columns show $((\text{Obj} - \text{Bound})/\text{Obj}) \times 100\%$. The "Group Size" codes (s = small, m = medium, l = large) denote the size category of each group, and the "Heterogeneity" codes (h = homogeneous, w = weakly heterogeneous, s = strongly heterogeneous) reflect item dimension variation within groups.

Items	Groups	Group Size	Heterogeneity	Obj	LB	GB-LB	LB Gap	GB-LB Gap
52	5	`mllll`	`swhhw`	11.3	2.16	9.37	80.80%	16.76%
45	5	`mllml`	`wwsss`	8.58	1.55	7.30	81.95%	14.94%
44	5	`mmlll`	`wwshh`	7.97	1.53	6.48	80.82%	18.70%
42	5	`slmll`	`shsws`	7.11	1.38	6.13	80.62%	13.76%
42	4	`lmll`	`hhhw`	6.31	1.40	5.06	77.87%	19.75%
41	4	`lmll`	`hwws`	7.23	1.49	5.87	79.36%	18.75%
40	5	`mllmm`	`sshws`	6.31	1.27	5.31	79.94%	15.89%
38	5	`lmlsl`	`hshhw`	5.24	1.11	4.41	78.90%	15.84%

Statistical Bound. Table 5 presents results for our three largest instances using the statistical lower bound approach described in Section 4.2.2. As detailed there, we fit a three-parameter Weibull distribution to a set of perturbed minima (Eqs. (26)–(28)) and construct a Golden–Alt confidence interval for the theoretical optimum. Table 5 presents the results for our three largest instances. The "Min" and "Max"columns shows the smallest and largest of the 100 observed minima, while the "KS (Stat, p-val)" and "AD (Stat, p-val)" columns confirm a satisfactory Weibull fit (none of the p-values are near conventional significance thresholds). The resulting Golden–Alt CI provides a statistical lower bound for each instance, with the upper bound set by the minimum observed value. Finally, the "Opt. Gap (%)" column compares our best-known feasible solution against that statistical lower bound. Overall, these results suggest that the best solutions are far closer to the global optimum than implied by either Gurobi's or our grid-based lower bound. While the procedure remains sensitive to sample size, distribution fit, and feasible minima generation, in practice, it provides a useful complementary perspective on solution quality—especially when deterministic bounds are weak. Next we shift focus from "how close to optimal?" to "which solution method performs best?" across all problem instances.

Table 5: Wilson's Statistical Lower Bound Results for the Three Largest Instances. Min, Max, and Golden-Alt CI values are scaled by 10^5.

Instance	Min	Max	KS (Stat, p-val)	AD (Stat, p-val)	Golden-Alt CI	Opt. Gap
42 items	6.31	6.74	(0.056/0.432)	(0.401/0.847)	(6.15, 6.31)	2.61%
45 items	8.58	9.31	(0.034/0.189)	(0.117/0.999)	(8.24, 8.58)	4.58%
52 items	11.3	12.0	(0.076/0.644)	(0.534/0.712)	(10.5, 11.3)	7.62%

5.2.4. Comparative Objective Performance

Figure 4 illustrates the percentage deviation of each solution technique's final objective value relative to the best objective value obtained for that instance, across all 90 problem instances. In the figure, solution approaches are ordered left to right by increasing median deviation. Solution approaches that start with "`win`" designate a sliding window approach with a set window size and per-subproblem time limit. For example, "`win_8it_60`" refers to an 8-item window, 60-second time limit, sliding window approach. As a benchmark, we use the single

MILP approach (labeled "`single`"): any method with median performance below that of single MILP is retained, while those that failed to improve on single MILP were excluded for clarity (e.g., sliding-window sizes of 3 or fewer and certain sliding-window combinations that struggled under 5-second time limits).

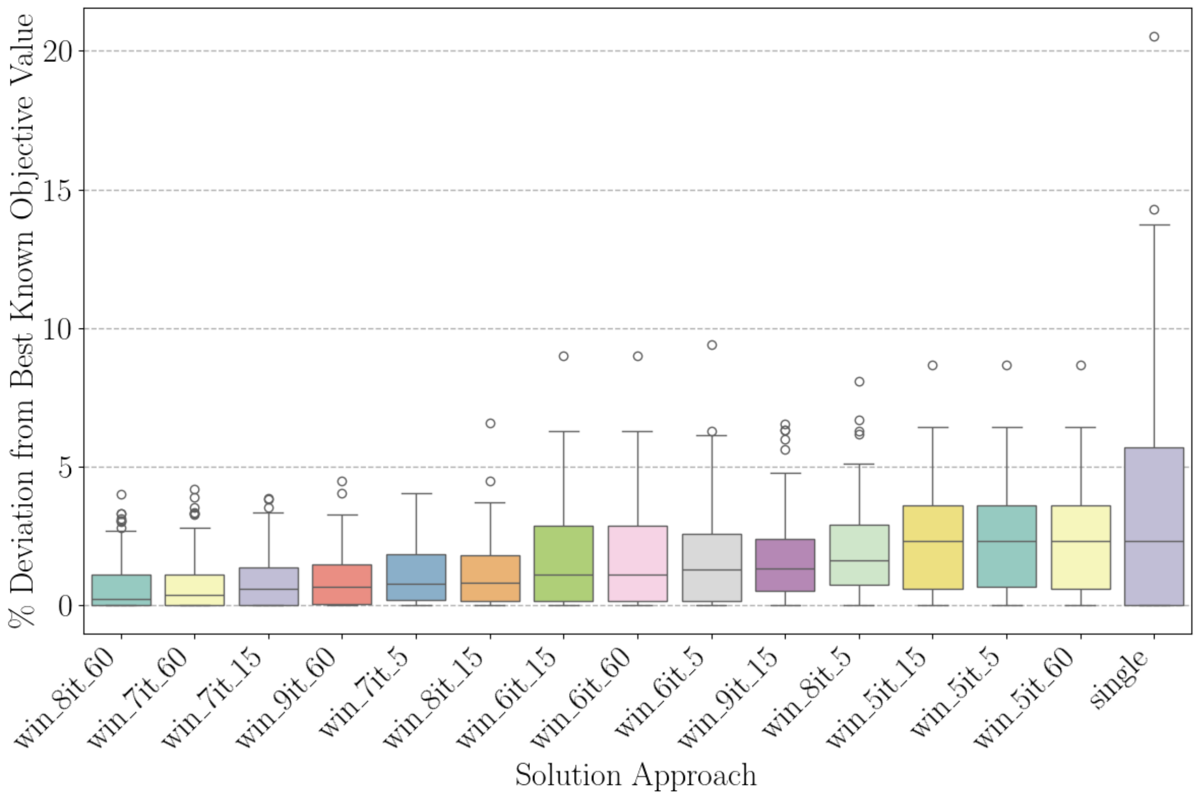

Figure 4: Boxplot of Percentage Deviation from Best Known Objective, Ordered by Increasing Median.

Decomposition methods consistently outperform the single MILP with visibly tighter deviation spreads, as seen in Figure 4. Allowing more time per window (e.g., 15 vs. 5 seconds) naturally reduces median deviation, but increasing the window size alone does not guarantee improvement. For example, a 9-item window with a 60-second limit ranks among the top performers with a median deviation less than 1%, yet its 9-item, 5-second variant performs worse than the single MILP and is not shown. The 8-item, 60-second sliding window method achieves the strongest median performance overall, although it requires an average total runtime of roughly 900 seconds (see Figure 3). By contrast, the most effective method at a 5-second time limit is the 7-item window, which finishes processing most instances in just over a minute while still outperforming single MILP.

These observations highlight the relationship between solution quality and computational effort. Figure 5 highlights this balance more explicitly. Here, each data point indicates a method's average percent deviation from best solution (y-axis) versus its average solve time (x-axis), across all problem instances. For single MILP, we record intermediate solver states at 100-second increments (though solver callbacks may align slightly off these intervals), then average the deviations. As the figure shows, decomposition methods define the Pareto front, with single MILP solutions lagging well behind. In particular, 5–7 item windows under a 5-

second limit appear to strike an excellent compromise between speed and solution quality, rarely exceeding a few percent deviation.

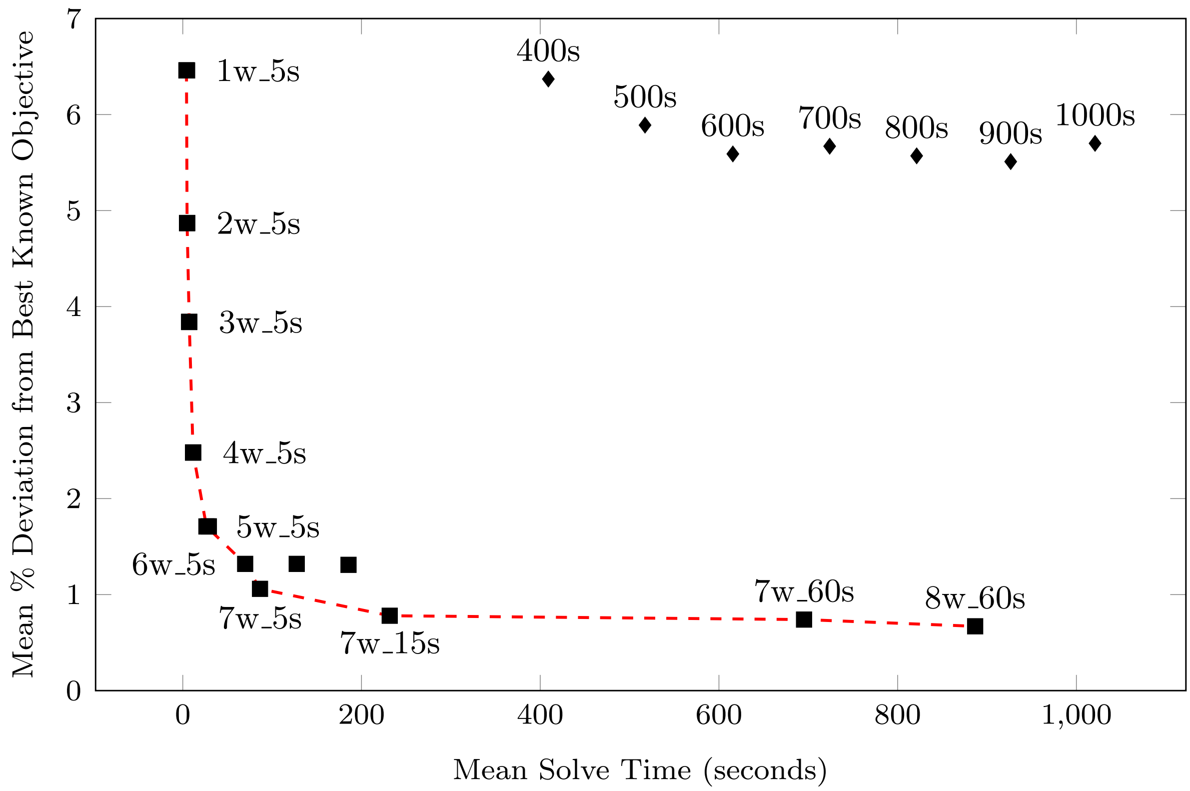

Figure 5: Pareto curve showing the relationship between mean solution time and mean percentage deviation from the best obtained objective values across all instances. Black squares represent sliding window decomposition solutions; black diamonds represent single MILP solutions recorded at various time intervals. The dashed line indicates the Pareto-efficient frontier, dominated by the decomposition methods.

Overall, the decomposition method significantly outperforms a single full-model solve in both time and quality. Larger window sizes can improve solution quality, but diminishing returns arise as subproblem size grows. Meanwhile, short time limits suffice for small windows, achieving excellent solutions in mere seconds or minutes. By choosing window sizes strategically, planners can solve large instances quickly while preserving high alignment with prioritization goals.

6. Conclusion

In this paper, we introduce a two-dimensional bin packing model that embeds multiple levels of prioritization into the objective, thereby unifying classical packing constraints with facility-layout-inspired distance preferences. Our approach centers on a prioritization matrix that captures both item-to-item adjacency needs and item-to-access-point proximity requirements. Through computational experiments, we demonstrate how this formulation can yield spatially dense configurations close to a bin access point while maintaining strong group unity and partial sequencing alignment when desired.

The proposed framework applies to combat loading in military logistics, warehouse and cargo stowage, multi-drop deliveries, or any application where fine-grained placement priorities influence overall layout. We show that single-level and multi-level prioritization can be accommodated by simply adjusting weights within a unified matrix, making the model highly flexible in handling diverse real-world scenarios, from roll-on/roll-off ship embarkation planning to automated pick-and-pack warehouses.

Practical Considerations. In implementing the two optimization-based methods discussed (single MILP and sliding window decomposition), we note several common trade-offs. First, proving optimality for even moderately sized instances can be prohibitively time-consuming, since bin packing and layout problems are known to exhibit exponential complexity (Meller et al., 1998). Consequently, imposing relatively short time limits on each subproblem in the sliding window approach often yields high-quality solutions without requiring the solver to close large optimality gaps. Second, seamless termination in the sliding-window method means each smaller subproblem finishes quickly, ensuring the overall procedure completes in a predictable manner. Although the decomposition strategy does not provide a global MILP gap, we observe that practical performance can be strong when decomposition parameters (e.g., subproblem sizes or time limits) are chosen judiciously.

Notably, both solution approaches are applicable whenever there is a well-defined bin access point and prioritized items or groups. If a future application requires purely intra-group clustering without a bin access point consideration, the group ordering logic would need to be revised to account for the lack of a distance-based pull toward a bin edge. Nonetheless, the broader concept of decomposing large layout problems into manageable subproblems still applies, illustrating the extensibility of our proposed framework to alternate objectives and constraints.

Future Directions. Our computational results demonstrate the effectiveness of the decomposition approach, offering planners a practical method to balance solution quality and runtime by tuning parameters (e.g., window size, time limits). Building on this, future algorithmic work could explore specialized heuristics or metaheuristics which may further reduce solve times or handle problem variants with additional constraints such as load balancing, non-adjacency requirements for hazardous items, or orientation restrictions (e.g., if certain cargo cannot be rotated).

Finally, the single-bin assumption could be lifted to investigate multi-bin or multi-vessel scenarios, incorporating a bin-selection phase that weighs how to distribute items among multiple bins while still respecting spatial priorities. Likewise, a scenario with multiple access points or varying item-level sequencing could be modeled by suitably extending the prioritization matrix. Overall, this study highlights the power of integrating spatial and priority-based considerations in bin packing. By illustrating a flexible distance-weighted formulation and showing how it can be efficiently solved via decomposition techniques, we provide a basis for future research in military, warehouse, and other logistics domains where both feasibility and strategic placement goals must be satisfied.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

This work was partially supported by the Omar N. Bradley Officer Research Fellowships in Mathematics.

References

Ahmadi, A., Pishvaee, M.S., Jokar, M.R.A., 2017. A survey on multi-floor facility layout problems. Computers & Industrial Engineering 107, 158–170. doi:10.1016/j.cie.2017.03.015.

Akyüz, M.H., Öncan, T., Altinel, I.K., 2010. The multi-commodity capacitated multi-facility weber problem: Heuristics and confidence intervals. IIE Transactions 42, 825–841. doi:10.1080/0740817X.2010.491504.

Altarazi, S., 2013. A new prioritizing-stacking heuristic algorithm for the inner-city truck loading problem. International Journal of Business and Management 8, 137. doi:10.5539/ijbm.v8n13p137.

Altinel, I.K., Aras, N., Oommen, B.J., 2000. Fast, efficient and accurate solutions to the hamiltonian path problem using neural approaches. Computers & Operations Research 27, 461–494. doi:10.1016/S0305-0548(99)00065-9.

Alvarez-Valdés, R., Parreño, F., Tamarit, J.M., 2013. A grasp/path relinking algorithm for two-and three-dimensional multiple bin-size bin packing problems. Computers & Operations Research 40, 3081–3090. doi:10.1016/j.cor.2013.07.015.

Bischoff, E.E., Ratcliff, M., 1995. Issues in the development of approaches to container loading. Omega 23, 377–390. doi:10.1016/0305-0483(95)00015-G.

Bortfeldt, A., Wäscher, G., 2013. Constraints in container loading – a state-of-the-art review. European Journal of Operational Research 229, 1–20. doi:10.1016/j.ejor.2012.12.006.

Bozer, Y.A., Meller, R.D., 1997. A reexamination of the distance-based facility layout problem. IIE Transactions 29, 549–560. doi:10.1080/07408179708966365.

Burkard, R.E., Rendl, F., 1984. A thermodynamically motivated simulation procedure for combinatorial optimization problems. European Journal of Operational Research 17, 169–174. doi:10.1016/0377-2217(84)90231-5.

Castro, P.M., Oliveira, J.F., 2011. Scheduling inspired models for two-dimensional packing problems. European Journal of Operational Research 215, 45–56. doi:10.1016/j.ejor.2011.06.001.

Chazelle, 1983. The bottom-left bin-packing heuristic: An efficient implementation. IEEE Transactions on Computers 100, 697–707. doi:10.1109/TC.1983.1676307.

Christensen, H.I., Khan, A., Pokutta, S., Tetali, P., 2017. Approximation and online algorithms for multidimensional bin packing: A survey. Computer Science Review 24, 63–79. doi:10.1016/j.cosrev.2016.12.001.

Coffman, E.G., Galambos, G., Martello, S., Vigo, D., 1999. Bin packing approximation algorithms: Combinatorial analysis, in: Handbook of Combinatorial Optimization: Supplement Volume A. Springer, pp. 151–207. doi:10.1007/978-1-4757-3023-4_3.

Delorme, M., Iori, M., Martello, S., 2016. Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research 255, 1–20. doi:10.1016/j.ejor.2016.04.030.

Derigs, U., 1985. Using confidence limits for the global optimum in combinatorial optimization. Operations Research 33, 1024–1049. doi:10.1287/opre.33.5.1024.

Dexterity AI, 2023. Dexterity AI and FedEx unveil first-of-its-kind robotics trailer loading technology. URL: https://www.dexterity.ai/blog/dexterity-ai-and-fedex-unveilfirst-of-its-kind-robotics-trailer-loading-technology/. accessed: 2025-01-30.

DHL, 2023. To optimize supply chains, optimize the warehouse. The Wall Street Journal URL: https://partners.wsj.com/blue-yonder/the-new-supply-chain/to-optimize-supply-chains-optimize-the-warehouse/. accessed: 2025-01-30.

Drira, A., Pierreval, H., Hajri-Gabouj, S., 2007. Facility layout problems: A survey. Annual Reviews in Control 31, 255–267. doi:10.1016/j.arcontrol.2007.04.001.

Erbayrak, S., Özkır, V., Yıldırım, U.M., 2021. Multi-objective 3d bin packing problem with load balance and product family concerns. Computers & Industrial Engineering 159, 107518. doi:10.1016/j.cie.2021.107518.

Filella, G.B., Trivella, A., Corman, F., 2023. Modeling soft unloading constraints in the multi-drop container loading problem. European Journal of Operational Research 308, 336–352. doi:10.1016/j.ejor.2022.10.033.

Fisher, R.A., Tippett, L.H.C., 1928. Limiting forms of the frequency distribution of the largest or smallest member of a sample, in: Mathematical proceedings of the Cambridge philosophical society, Cambridge University Press. pp. 180–190. doi:10.1017/S0305004100015681.

Fontaine, P., Minner, S., 2023. A branch-and-repair method for three-dimensional bin selection and packing in e-commerce. Operations Research 71, 273–288. doi:10.1287/opre.2022.2369.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco.

General Dynamics Land Systems, 2025. M1A2 Abrams Main Battle Tank Specifications. https://international.gdls.com/english/products/ABRAMS/M1A2.pdf. Accessed: 2025-03-01.

Golden, B.L., Alt, F.B., 1979. Interval estimation of a global optimum for large combinatorial problems. Naval Research Logistics Quarterly 26, 69–77. doi:10.1002/nav.3800260108.

Goodman, S., Pohl, J.G., 2003. ICODES: A Ship Load-Planning System, in: Proceedings of the 13th International Ship Control Systems Symposium (SCSS), Orlando, FL. pp. 1–10.

Grosse, E.H., Glock, C.H., Neumann, W.P., 2017. Human factors in order picking: a content analysis of the literature. International Journal of Production Research 55, 1260–1276. doi:10.1080/00207543.2016.1186296.

Gu, J., Goetschalckx, M., McGinnis, L.F., 2007. Research on warehouse operation: A comprehensive review. European Journal of Operational Research 177, 1–21. doi:10.1016/j.ejor.2006.02.025.

Hopper, E., Turton, B.C., 2001. An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. European Journal of Operational Research 128, 34–57. doi:10.1016/S0377-2217(99)00357-4.

Hübner, A., Holzapfel, A., Kuhn, H., 2015. Operations management in multi-channel retailing: an exploratory study. Operations Management Research 8, 84–100. doi:10.1007/s12063-015-0101-9.

Iori, M., De Lima, V.L., Martello, S., Miyazawa, F.K., Monaci, M., 2021. Exact solution techniques for two-dimensional cutting and packing. European Journal of Operational Research 289, 399–415. doi:10.1016/j.ejor.2020.06.050.

Joint Chiefs of Staff, 2019. Joint Publication 3-02: Amphibious Operations. United States Department of Defense. URL: https://www.jcs.mil/Doctrine/Joint-Doctrine-Pubs/3-0-Operations-Series/.

Kirschenman, W., McConnell, B., King, R., 2024. Automated vessel selection and combat load planning. Army Sustainment 56, 58–61. https://www.lib.ncsu.edu/resolver/1840.20/44301.

Koopmans, T.C., Beckmann, M., 1957. Assignment problems and the location of economic activities. Econometrica 25, 53–76. doi:10.2307/1907742.

Li, X., Zhang, K., 2018. Single batch processing machine scheduling with two-dimensional bin packing constraints. International Journal of Production Economics 196, 113–121. doi:10.1016/j.ijpe.2017.11.015.

Lodi, A., Martello, S., Monaci, M., 2002. Two-dimensional packing problems: A survey. European Journal of Operational Research 141, 241–252. doi:10.1016/S0377-2217(02)00123-6.

Lodi, A., Martello, S., Vigo, D., 1999. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. INFORMS Journal on Computing 11, 345–357. doi:10.1287/ijoc.11.4.345.

Meller, R.D., Narayanan, V., Vance, P.H., 1998. Optimal facility layout design. Operations Research Letters 23, 117–127. doi:10.1016/S0167-6377(98)00024-8.

do Nascimento, O.X., de Queiroz, T.A., Junqueira, L., 2021. Practical constraints in the container loading problem: Comprehensive formulations and exact algorithm. Computers & Operations Research 128, 105186. doi:10.1016/j.cor.2020.105186.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. The Computer Journal 7, 308–313. doi:10.1093/comjnl/7.4.308.

Olist, 2019. Brazilian e-commerce public data set by Olist. https://www.kaggle.com/olistbr/brazilianecommerce. Accessed: 2025-03-27.

Paquay, C., Schyns, M., Limbourg, S., 2016. A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. International Transactions in Operational Research 23, 187–213. doi:10.1111/itor.12111.

Tate, D.M., Smith, A.E., 1995. A genetic approach to the quadratic assignment problem. Computers & Operations Research 22, 73–83. doi:10.1016/0305-0548(93)E0020-T.

Trivella, A., Pisinger, D., 2016. The load-balanced multi-dimensional bin-packing problem. Computers & Operations Research 74, 152–164. doi:10.1016/j.cor.2016.04.020.

United States Navy, 2021. Large, Medium-Speed, Roll-on/Roll-off Ships T-AKR Fact File. https://www.navy.mil/Resources/Fact-Files/Display-FactFiles/Article/2226172/large-medium-speed-roll-onroll-off-ships-t-akr/. Accessed: 2025-03-01.

United States Navy, 2023. Landing Craft, Air Cushion (LCAC) Fact File. https://www.navy.mil/Resources/Fact-Files/Display-FactFiles/Article/2170004/landing-craft-air-cushion-lcac/. Accessed: 2025-03-01.

UPS, 2023. Warehouse optimization: A guide to improving efficiency and reducing costs. URL: https://www.ups.com/assets/resources/webcontent/UPS_Customer_Solutions_Warehouse_Optimization.pdf/. accessed: 2025-01-30.

Vancroonenburg, W., Verstichel, J., Tavernier, K., Berghe, G.V., 2014. Automatic air cargo selection and weight balancing: a mixed integer programming approach. Transportation Research Part E: Logistics and Transportation Review 65, 70–83. doi:10.1016/j.tre.2013.12.013.

Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. European Journal of Operational Research 183, 1109–1130. doi:10.1016/j.ejor.2005.12.047.

Wilson, A.D., King, R.E., Wilson, J.R., 2004. Case study on statistically estimating minimum makespan for flow line scheduling problems. European Journal of Operational Research 155, 439–454. doi:10.1016/S0377-2217(02)00910-4.

X Corps Headquarters, 1951. G-4 Summary, 15 Aug to 30 Sep 1950, in: X Corps War Diary Summary for Operation Chromite, 15 August to 30 September 1950, p. 40. Section IV within Part II of unpublished official record. Year inferred from related map dated 8 June 1951. Pagination likely relative to start of Part II. Approved by LTG E. M. Almond and COL J. S. Guthrie.

Zanakis, S.H., 1979. A simulation study of some simple estimators for the three-parameter weibull distribution. Journal of Statistical Computation and Simulation 9, 101–116. doi:10.1080/00949657908810302.