


Workload Balancing in a Food Distribution Center: Comparing Constraint Programming and Mixed-Integer Linear Programming

Josep Maria Salvia Hornos¹ ✉ 

Department of Computer Engineering and Digital Design, University of Lleida, Spain

Department of IT and Systems, Artificial Intelligence Division, Corporación Alimentaria Guissona S.A., Spain

Cèsar Fernández Camón ✉ 

Department of Computer Engineering and Digital Design, University of Lleida, Spain

Carles Mateu Piñol ✉ 

Department of Computer Engineering and Digital Design, University of Lleida, Spain

Ivet Rafegas Fonoll ✉

Department of IT and Systems, Artificial Intelligence Division, Corporación Alimentaria Guissona S.A., Spain

Abstract

Operational imbalances in high-volume food manufacturing warehouses frequently lead to critical bottlenecks, diminishing overall throughput and efficiency. This paper presents Constraint Programming (CP) and Mixed-Integer Linear Programming (MILP) models designed to rebalance ingestion workloads by optimally assigning articles to input machines, reconfiguring an existing warehouse. The model considers aggregated workload over discretized time periods and incorporates practical constraints to ensure operational feasibility. Applied to data from a central storage facility of one of Spain's largest food manufacturers that handles facility multimillion box per year. In this paper our approach demonstrates a potential 30% improvement in workload balance with approximately 10% of slot reconfigurations. This proactive rebalancing offers a robust strategy for mitigating congestion and enhancing the efficiency of critical food supply chain nodes.

2012 ACM Subject Classification Applied computing → Operations research; Applied computing → Supply chain management

Keywords and phrases Workload balancing, Constraint Programming (CP), Mixed-Integer Linear Programming (MILP), Warehousing, Bottleneck mitigation, CP-SAT, SCIP, OR-Tools

Digital Object Identifier 10.4230/LIPIcs.CP.2025

Funding *Josep Maria Salvia Hornos*: this work was partially funded by Generalitat de Catalunya AGAUR DI-2024-00020

Cèsar Fernández Camón: reports funding by the Spanish MCIN/AEI/10.13039/501100011033/, FEDER, UE in project IDs PID2022-138564OA-I00 and PID2022-137971OB-I00

Carles Mateu Piñol: this work was partially funded by the Ministerio de Ciencia e Innovación - Agencia Estatal de Investigación (AEI) (PID2021-123511OB-C31- MCIN/AEI/ 10.13039/501100011033/FEDER, UE and RED2022-134219-T)

1 Introduction

Warehousing is a central piece of modern supply chains, providing essential buffering capacity, allowing asynchrony between production and dispatch schedules, enabling efficient order

¹ Corresponding Author



fulfillment, and supporting diverse manufacturing and distribution strategies. The increasing adoption of automation in warehouse facilities has introduced new levels of efficiency and speed, but also new operational complexities [4]. Efficient warehouse operations are important in modern supply chains, particularly for high volume facilities like those in the food manufacturing sector [11].

Extensive research has been dedicated to challenges such as the Storage Location Assignment Problem (SLAP), which aims to determine optimal product placements to minimize travel times and restocking costs, as reviewed by [5]. Approaches to SLAP vary widely, from traditional integer programming (IP) to more recent applications of Constraint Programming (CP), which has demonstrated significant advantages in finding solutions for large, diverse product instances [12]. Furthermore, adaptive AI-driven algorithms have been developed for SLAP to handle multiple, potentially non-linear objectives [13].

Beyond storage assignment, other operational facets significantly impact warehouse throughput. For instance, Constraint Programming has also been successfully applied to optimize stochastic inventory systems, managing uncertainties in demand and costs [10], and to enhance energy efficiency alongside time performance in automated storage and retrieval systems (AS/RS [9, 3]) through advanced control policies [6]. While these areas address critical aspects of product storage and retrieval, the initial phase of product ingestion and its distribution among processing units presents a distinct set of challenges. Specifically, imbalances in workload assigned to input machines can create severe bottlenecks that propagate through the system, a problem particularly acute in high-throughput environments like the one motivating this study. Our work addresses this specific challenge of minimizing operational imbalances by strategically assigning tasks to input machines.

This paper addresses the problem of minimizing such operational imbalances by strategically assigning tasks to input machines. Unlike approaches that rely on simulation-based optimization (derivative-free, black box systems), or AI-driven heuristics suitable for highly non-linear problems such as adaptive SLAP [13], we propose a simplified model-based approach using Constraint Programming (CP) and Mixed-Integer Linear Programming (MILP). The core idea is that the total amount of work assigned to each machine serves as a direct proxy for its potential to become a bottleneck. This way, we can proactively mitigate congestion by balancing this workload across all input machines.

2 Mathematical Model for Ingestion Workload Balancing

This section details the models developed to balance ingestion workload. The primary goal is to assign articles to input machines such that the workload is as equitable as possible for each transportation lane and input machine over given time periods, minimizing potential bottlenecks in the buffers. We will first explain the input mechanism and assumptions before introducing the formalization.

2.1 Preliminaries of the input mechanism

The warehousing system processes incoming stock units (boxes), each of a specific article type. The journey of these boxes from production to storage involves several stages:

1. Boxes are produced at different points in time from different points of the factory and place onto transportation lanes.
2. Boxes are assigned to a specific FIFO storage lane. This process is highly dependent on available storage, current workload and other variables. For the purposes of this

discussion, we can consider this process stochastic.

3. To reach the storage lane, the box has to be placed by an input machine; and to be picked up by the input machine, it must first be placed on an associated buffer, of which there is one buffer per transportation lane and input machine. So, to reach the storage lane, the box is automatically routed to the corresponding buffer.
4. Input machines pick boxes from their buffers and place them into the storage lanes. Each input machine has storage matrices associated to it, where each cell of the matrix is a storage lane. Each storage matrix is assigned to only one input machine.

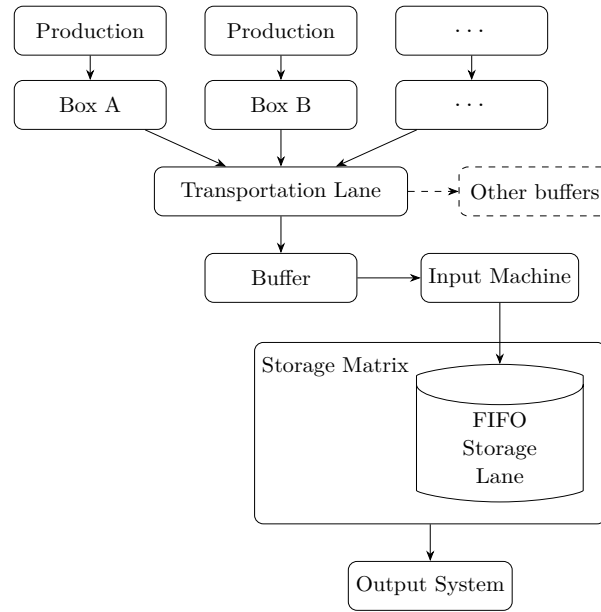


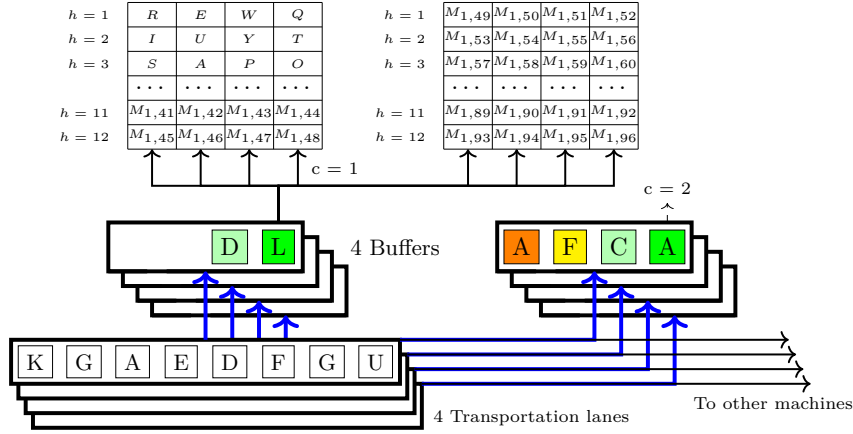
Figure 1 End-to-end flow from dual production lines to output, highlighting the FIFO Storage Lane within the Storage Matrix.

A schematic of this flow is shown in Figure 1.

The configuration of article assignments within the storage matrix directly influences buffer workload distribution. This dependency emerges from two factors: (1) articles exhibit heterogeneous production rates and packaging densities, generating variable pressure on the buffers; (2) temporal production schedules and spatial distribution across the factory create distinct transportation pathways that affect buffer utilization through transportation lane work unbalance.

The most critical point of failure of this system is the buffers, which have small capacities. A full buffer can cause incoming boxes for that buffer to block the main transportation lane, blocking flow to other input machines downstream on the same lane. This is illustrated in Figure 2. Where one of the buffer assigned to input machine $c = 2$ is full, which could potentially cause a blockage.

Balanced workload distribution prevents system failure propagation. When machines require maintenance or experience errors imbalanced systems exhibit higher failure rates due to workload concentration. Balanced configurations reduce overload probability by distributing excess capacity across multiple buffers, enabling fault tolerance by trying to prevent exceeding buffer limits.



■ **Figure 2** Depiction of two storage matrices of 12 x 4 storage lanes of one input machine ($c = 1$), with its corresponding 4 buffers. Input machine ($c = 2$) partially depicted with the four buffers. There are 96 storage lanes depicted, going from $M_{1,1} = R$ to $M_{1,96}$, product types (R, E, W, Q, L, D, \dots) are provided as examples.

110 2.2 Model assumptions and abstractions

111 The formulation employs several abstractions that balance computational tractability with
 112 capturing warehouse dynamics for relocation strategies. Ideal models would be able to
 113 consider individual events, exact time blocks, and storage capacities. Such precise models
 114 were tried by the authors but were deemed intractable and inefficient, both because of their
 115 computational complexity and their difficulty generalizing to unseen circumstances.

116 Experimentally, we have seen that when multiple input machines handle the same product
 117 type, incoming workload from each lane distributes roughly equally among them, so we take
 118 this as our basic modeling assumption. Our approach deals instead with discretized time
 119 blocks, where the workloads of the machines are treated as continuous, aggregated quantities
 120 distributed across product types and processing lanes. The model will therefore focus on the
 121 combinatorial challenge of determining which sets of input machines should handle which
 122 product types.

123 The model deliberately avoids explicit representation of physical warehouse layout,
 124 machine travel times, detailed buffer dynamics beyond their influence on lane workload
 125 requirements, or complex storage routing decisions; we leave this optimization to the next
 126 level of the hierarchy responsible for handling execution.

127 The model operates within predefined warehousing constraints: each product type
 128 must be assigned to a specific number of input machines, and each machine must handle a
 129 predetermined number of different products. These constraints acknowledge practical realities
 130 where complete system redesigns are often infeasible. For instance, a product might always
 131 require three input machines for adequate processing capacity, but the model determines
 132 which specific three machines provide the best overall system performance. This significantly
 133 reduces the search space while ensuring proposed solutions align with realistic operational
 134 change objectives. Moreover, certain extra considerations are included in the formulation,
 135 like blocking certain items or slots from being reordered.

2.2.1 Sets, indices, maps, parameters and decision variables

In this subsection, we introduce the sets, indices, and mappings used to define the model structure, the parameters that capture input data and system constraints, and the decision variables that will be optimized.

Sets, indices and maps

- \mathcal{A} : Set of unique articles, indexed by a .
- \mathcal{X} : Set of unique articles ($\mathcal{X} \subseteq \mathcal{A}$) that cannot be reordered, indexed by x .
- \mathcal{C} : Set of input machines indexed by c .
- \mathcal{T} : Set of discrete time periods under consideration, indexed by t .
- \mathcal{L} : Set of transportation lanes, indexed by l .
- \mathcal{I}_c : Set of storage slots of the matrices associated with machine $c \in \mathcal{C}$, indexed by j .
- Ψ : The set of all addressable machine-slot pairs, $\Psi = \{(c, j) \mid c \in \mathcal{C}, j \in \mathcal{I}_c\}$.
- $M : \Psi \rightarrow \mathcal{A}$. In expressions, $M_{c,j}$ denotes the value $M(c, j) \in \mathcal{A}$, which is the article assigned to the slot $j \in \mathcal{I}_c$ of input machine $c \in \mathcal{C}$.

By this formulation, the number of buffers in the system is $|\mathcal{L}| \cdot |\mathcal{C}|$.

Parameters

- $O_{a,l,t}$: Total number of boxes associated with article a arriving in time period t through transportation lane l . This is a non-negative real number.
- $G_{a,c}$: A binary parameter that equals 1 if article a is assigned to input machine c and 0 otherwise. This represents the existing assignment configuration. Algebraically, $G_{a,c} = 1 \iff \exists j \in \mathcal{I}_c$ such that $M_{c,j} = a$; otherwise, $G_{a,c} = 0$.
- N_a : The number of distinct input machines to which article a is assigned in the map G . This is a positive integer constant for each article a , calculated as $N_a = \sum_{c \in \mathcal{C}} G_{a,c}$.
- N_c : The number of distinct articles assigned to input machine c in the map G . This is a positive integer constant for each input machine c , calculated as $N_c = \sum_{a \in \mathcal{A}} G_{a,c}$.
- $R_{a,l,t}$: Per-input machine workload for article a in time period t on transportation lane l . This is calculated as $R_{a,l,t} = O_{a,l,t}/N_a$, assuming the total workload $O_{a,l,t}$ for an article is evenly distributed among the N_a input machines. This is a non-negative real number.
- λ : Scale factor used for integer approximation in CP encoding. In MILP, $\lambda = 1$.
- $\bar{W}_{l,t}$: Average workload in time period t through transportation lane l , computed as $\bar{W}_{l,t} = \frac{\lambda}{|\mathcal{C}|} \sum_{a \in \mathcal{A}} O_{a,l,t}$. This value serves as a target for balanced workload distribution and is equivalent to the actual average workload $\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} W_{c,l,t}$. This is a non-negative real number in MILP, and a positive integer in CP.
- K : maximum number of allowed changes between the current configuration (G) and the new configuration. This is a positive integer.

Decision Variables

- $S_{a,c}$: A binary variable that equals 1 if article a is assigned to be handled by input machine c , and 0 otherwise.
- $W_{c,l,t}$: A non-negative (continuous in MILP; integer in CP) variable representing the total workload assigned to input machine c in time period t for transportation lane l .
- $E_{c,l,t}$: A non-negative continuous (continuous in MILP; integer in CP) variable representing the deviation (or error) of machine c workload from the target workload ($\bar{W}_{l,t}$) in time period t for transportation lane l .

179 The domains of these variables are:

$$180 \quad S_{a,c} \in \{0, 1\} \quad \forall a \in \mathcal{A}, \forall c \in \mathcal{C} \quad (1)$$

$$181 \quad W_{c,l,t} \geq 0 \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \quad (2)$$

$$182 \quad E_{c,l,t} \geq 0 \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \quad (3)$$

183 2.3 Constraints & objective

184 The objective is to minimize the sum of absolute deviations of each input machine's workload
185 from the target average workload, across all input machines and all time periods. This
186 promotes an equitable distribution of work.

$$187 \quad \min \sum_{c \in \mathcal{C}} \sum_{l \in \mathcal{L}} \sum_{t \in \mathcal{T}} E_{c,l,t} \quad (4)$$

188 The model's assignments and workload calculations are governed by the following con-
189 straints.

190 **Workload Calculation for input machines:** The total workload $W_{c,t}$ for each input
191 machine c in each time period t is the sum of the per-input machine workloads $R_{a,t}$ for all
192 articles a assigned to it:

$$193 \quad W_{c,l,t} = \sum_{a \in \mathcal{A}} S_{a,c} \cdot R_{a,l,t} \cdot \lambda \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \quad (5)$$

194 **Deviation Calculation:** The deviation variable $E_{c,t}$ must capture the absolute difference
195 between the actual workload $W_{c,t}$ and the target average workload $\bar{W}_{l,t}$. These constraints
196 ensure $E_{c,l,t} = |W_{c,l,t} - \bar{W}_{l,t}|$. We linearize the constraints as follows:

$$197 \quad E_{c,l,t} \geq W_{c,l,t} - \bar{W}_{l,t} \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \quad (6)$$

$$198 \quad E_{c,l,t} \geq \bar{W}_{l,t} - W_{c,l,t} \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T}, \forall l \in \mathcal{L} \quad (7)$$

199 **Assignment Constraints:** These constraints ensure that the assignment rules are met:

200 1. Each article a must be assigned to exactly N_a distinct input machines:

$$201 \quad \sum_{c \in \mathcal{C}} S_{a,c} = N_a \quad \forall a \in \mathcal{A} \quad (8)$$

202 2. Each input machine c must be assigned exactly N_c distinct articles:

$$203 \quad \sum_{a \in \mathcal{A}} S_{a,c} = N_c \quad \forall c \in \mathcal{C} \quad (9)$$

204 Since both parameters N_a and N_c are derived from an existing valid solution G , this ensures
205 that the problem is feasible satisfying Equation (8) and (9) exists. Specifically, the total
206 number of assignment slots must be equal: $\sum_{a \in \mathcal{A}} N_a = \sum_{c \in \mathcal{C}} N_c$.

207 **Non reassignment policy:** Some articles must not be reassigned due to warehouse
208 logistics constraints:

$$209 \quad S_{x,c} = 1 \quad \forall c \in \mathcal{C}, \forall x \in \mathcal{X} \quad (10)$$

210 **Maximum number of changes:** The resulting configuration must have at most K changes,
211 which is equivalent to saying that $\sum_{c \in \mathcal{C}} \sum_{a \in \mathcal{A}} G_{a,c} \oplus S_{a,c} \leq K$:

$$212 \quad \sum_{c \in \mathcal{C}} \sum_{a \in \mathcal{A} \text{ s.t. } G_{a,c}=1} S_{a,c} \geq -K + \sum_{c \in \mathcal{C}} N_c \quad (11)$$

3 Experimental evaluation

We investigate the trade-offs between solution quality, the extent of allowed changes from an initial configuration, and the computational effort required. To test this, we selected 25 days of production as historical data and 15 future days as validation data out of real-world data from April and May of 2025. The reason for requiring a set of validation data is that the articles/time distribution is not guaranteed to be stable over time, and may change as the factory evolves, therefore, we want to ensure optimization on past historical data corresponds to improvements in the future. We discretized time intervals in groups of 2 hours which can approximately fit production patterns throughout the day. We also filtered these intervals, ensuring only periods when bottlenecks usually happen to be considered. The MILP problems were solved with SCIP [2, 1] and the CP version was solved with OR-Tools [8, 7], both were stopped at 1% optimality gap. For the CP version, λ was tested at many different values; for visual clarity, we picked $\lambda = 2$ and $\lambda = 10$ as representatives.

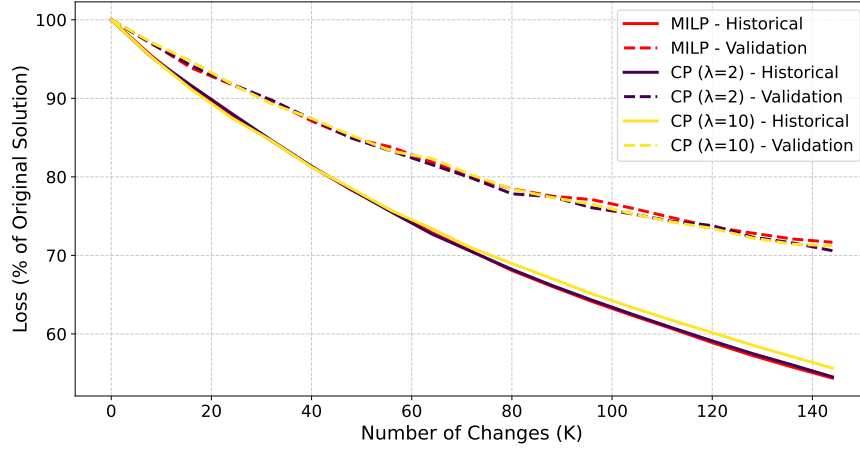


Figure 3 % Improvement as a function of K-distance for CP and MILP models based on historical data and validation data.

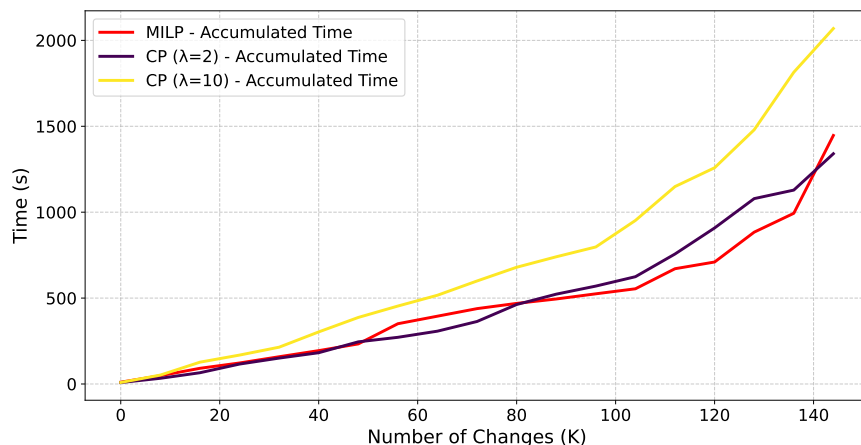
For reference of the experiment, $|\mathcal{A}| = 676$, $|\mathcal{X}| = 286$, $|\mathcal{C}| = 15$, $|\mathcal{L}| = 4$, for historical: $|\mathcal{T}| = 29$ and for validation: $|\mathcal{T}| = 27$.

In the following graphs, a K-distance of 0 zero changes corresponds to the initial assignment, where the changes correspond to the accumulated degrees of freedom in the article assignment ($S_{a,c}$). Figure 3 shows how the loss varies as the number of allowed changes (K) increases. Both approaches, using CP and using MILP, are shown on the graph. The x-axis shows the accumulated K-distance, and the y-axis shows the corresponding improvement loss. The improvement is computed with respect to the original configuration in the objective function (0 steps). Historical and validation are differentiated with solid and dashed lines respectively. Interestingly, the CP approximation version equivalent performance levels to the MILP version with a very low value of λ .

As the accumulated number of changes K increases, the historical loss demonstrates a corresponding improvement, with the cost function achieving further reduction due to the increased degrees of freedom available for optimization. This enhanced performance on the historical data is accompanied by improvements on the validation data, though the magnitude of validation set improvement is more modest compared to the historical improvements. The absence of performance plateaus on the validation set suggests that the current optimization

has not yet reached the limits of achievable performance.

At the tested simulations, we achieve a 30% improvement in balance buffer workload allocation with around 140 to 150 changes. This is significant, since the warehouse is composed of approximately 1500 slots, suggesting that at early stages a significant improvement in rebalancing may bring substantial long-term efficiency gains.



■ **Figure 4** Cumulative Solution Time as a function of K -distance for both MILP and CP models

This analysis examines the computational time requirements of different optimization steps as a function of the target K -distance. Figure 4 presents a comparative evaluation of cumulative solution time versus K -distance for absolute optimization. and various stepped optimization approaches, where the x-axis represents the total K -distance achieved and the y-axis depicts the cumulative solution time. The presented CP approach performs slightly worse than the MILP version at $\lambda = 10$, but offer a very similar performance profile at $\lambda = 2$, even though a perfect approximation of MILP could only be achieved with $\lambda \rightarrow \infty$.

We terminated the experiments at a distance of 150, where the solution time begins to exhibit exponential growth characteristics. This phenomenon can be attributed to the underlying structure: while proving optimality for highly disordered warehouse configurations is computationally tractable, the complexity increases dramatically as the warehouse configuration approaches more organized states.

4 Conclusions

The proposed CP and MILP workload balancing models achieve a 30% improvement in balance buffer workload allocation with $\approx 10\%$ changes in the slots. With similar performance at a well tuned λ parameter. Historical improvements consistently translate to meaningful performance gains on unseen data, with no observed performance plateaus indicating potential for further enhancement. Solution time exhibits exponential growth at certain deviations from the original configuration, reflecting the complexity transition from disordered to organized configurations. Larger step sizes provide early computational advantages but incur higher penalties in the exponential curve. Future work should integrate predictive modeling to address seasonal variations and changing workload patterns and should integrate regularization techniques to enhance solution robustness and adaptability.

References

- 1 Tobias Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1:1–41, 2009.
- 2 Suresh Bolusani, Mathieu Besançon, Ksenia Bestuzheva, Antonia Chmiela, João Dionísio, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Mohammed Ghannam, Ambros Gleixner, et al. The SCIP optimization suite 9.0. *arXiv preprint arXiv:2402.17702*, 2024.
- 3 Jean-Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. Models for automated storage and retrieval systems: a literature review. *International Journal of Production Research*, 50(24):7110–7125, 2012.
- 4 Jan Karasek. An overview of warehouse optimization. *International journal of advances in telecommunications, electrotechnics, signals and systems*, 2(3):111–117, 2013.
- 5 Lucy Medrano-Zarazúa, Jania Astrid Saucedo-Martínez, and Johanna Bolaños-Zuñiga. Storage location assignment problem in a warehouse: A literature review. In *International Conference on Computer Science and Health Engineering*, pages 15–37. Springer, 2022.
- 6 Antonella Meneghetti, Eleonora Dal Borgo, and Luca Monti. Rack shape and energy efficient operations in automated storage and retrieval systems. *International journal of production research*, 53(23):7090–7103, 2015.
- 7 Laurent Perron, Frédéric Didier, and Steven Gay. The CP-SAT-LP solver. In Roland H. C. Yap, editor, *29th International Conference on Principles and Practice of Constraint Programming (CP 2023)*, volume 280 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:2, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/opus/volltexte/2023/19040>, doi:10.4230/LIPIcs.CP.2023.3.
- 8 Laurent Perron and Frédéric Didier. CP-SAT. URL: https://developers.google.com/optimization/cp/cp_solver/.
- 9 Kees Jan Roodbergen and Iris FA Vis. A survey of literature on automated storage and retrieval systems. *European journal of operational research*, 194(2):343–362, 2009.
- 10 Roberto Rossi, S Armagan Tarim, Brahim Hnich, and Steven Prestwich. Constraint programming for stochastic inventory systems under shortage cost. *Annals of Operations Research*, 195:49–71, 2012.
- 11 Fabio Sgarbossa, Anita Romsdal, Olumide Emmanuel Oluyisola, and Jan Ola Strandhagen. Digitalization in production and warehousing in food supply chains. In *The Digital Supply Chain*, pages 273–287. Elsevier, 2022.
- 12 Claudio Telha and Rosa G González-Ramírez. A constraint-programming approach for the storage space allocation problem in a distribution center. *International Transactions in Operational Research*, 2025.
- 13 Amir Zarinchang, Kevin Lee, Iman Avazpour, Jun Yang, Dongxing Zhang, and George K Knopf. Adaptive warehouse storage location assignment with considerations to order-picking efficiency and worker safety. *Journal of Industrial and Production Engineering*, 41(1):40–59, 2024.