

## Adaptive warehouse storage location assignment with considerations to order-picking efficiency and worker safety

Amir Zarinchang, Kevin Lee, Iman Avazpour, Jun Yang, Dongxing Zhang & George K. Knopf

To cite this article: Amir Zarinchang, Kevin Lee, Iman Avazpour, Jun Yang, Dongxing Zhang & George K. Knopf (2024) Adaptive warehouse storage location assignment with considerations to order-picking efficiency and worker safety, Journal of Industrial and Production Engineering, 41:1, 40-59, DOI: [10.1080/21681015.2023.2263009](https://doi.org/10.1080/21681015.2023.2263009)

To link to this article: <https://doi.org/10.1080/21681015.2023.2263009>



© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 07 Jan 2024.



Submit your article to this journal [↗](#)



Article views: 536



View related articles [↗](#)



View Crossmark data [↗](#)

# Adaptive warehouse storage location assignment with considerations to order-picking efficiency and worker safety

Amir Zarinchang<sup>a</sup>, Kevin Lee<sup>b</sup>, Iman Avazpour<sup>c</sup>, Jun Yang<sup>a,d</sup>, Dongxing Zhang<sup>d</sup> and George K. Knopf<sup>a</sup>

<sup>a</sup>Department of Mechanical and Materials Engineering, The University of Western Ontario, London, Ontario, Canada; <sup>b</sup>School of Information Technology, Deakin University, Melbourne, Victoria, Australia; <sup>c</sup>Enabling Technology Solutions Pty. Ltd., Melbourne, Victoria, Australia; <sup>d</sup>Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China

## ABSTRACT

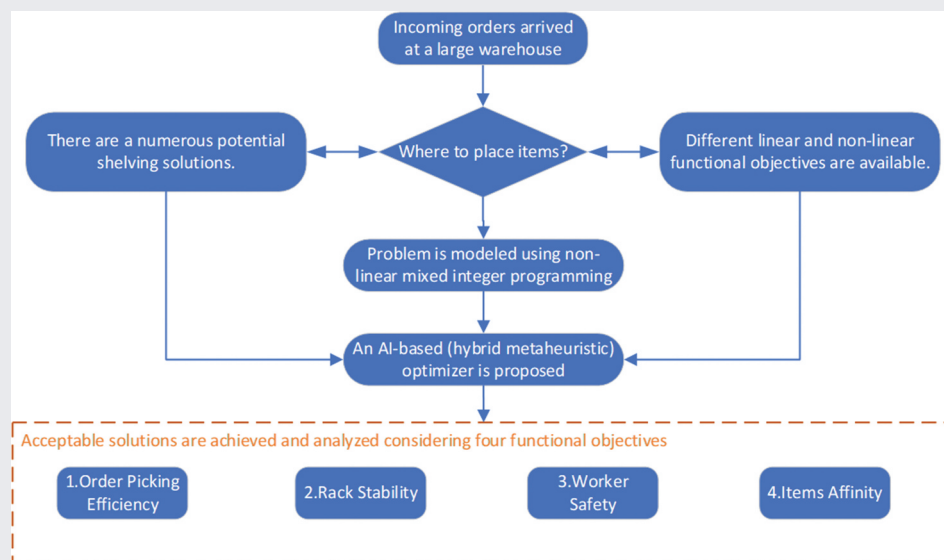
Smart warehouses require software-based decision-making tools to manage the receiving, storing, and picking of products. A major challenge in achieving efficient operations is deciding where to store products associated with incoming orders. The storage location assignment problem (SLAP) is more complex in large-size warehouses due to several functional objectives and numerous possible shelving solutions. This paper introduces an artificial intelligence algorithm that seeks to find an acceptable solution to SLAP with presented linear and nonlinear objective functions. The near-optimal technique exploits basin-hopping and simulated-annealing algorithms to find a solution when considering four functional objectives including worker safety, which has not been optimized using similar approaches. The algorithm is experimentally evaluated, and results demonstrate that reasonably achieved solutions are comparable to those obtained by well-known existing solvers. Furthermore, the problem could be solved with non-linear objectives which is beyond the commercial solvers' like SCIP capability.

## ARTICLE HISTORY

Received 19 May 2023  
Revised 19 September 2023  
Accepted 19 September 2023

## KEYWORDS



Storage location assignment problem (SLAP); order-picking efficiency; worker safety; nonlinear assignment problems; basin-hopping; simulated annealing



## 1. Introduction

Material handling operations in a warehouse often require a costly decision-making process to manage the receiving, storing, picking, sorting, and shipping of products. In general, retailers and wholesalers around the world receive palletized items of varying size, weight, and volume, keep them for a set period and then retrieve and ship them according to customer demands. The unique items in a pallet are also called *Stock Keeping Units* (SKUs) which are the smallest units that can be sold. The locations of the received pallets or SKUs in the warehouse are

determined by a warehouse management system. This is a complex problem involving several objectives, technical and safety constraints, and allocation criteria like item weight and product popularity [1–3]. One of the major aims of warehouse operations systems is to decide where to store the incoming orders to help facilitate the order-picking process for item retrieval. This is known as *storage location assignment problem* (SLAP) [1]. As the number of stored SKUs increases in the facility, the number of possible shelving locations becomes very large. For example, a small incoming order that contains 20 pallets in

**CONTACT** George K. Knopf  [gkknopf@uwo.ca](mailto:gkknopf@uwo.ca)  Department of Mechanical and Materials Engineering, The University of Western Ontario, London, Ontario, Canada

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

a warehouse with 100 available storage locations would result in  $1.3039 \times 10^{39}$  unique solutions if every location in the warehouse can accommodate one pallet with no allocation constraints.

These industrial warehouses are increasingly utilizing Industry 4.0 technologies in their logistics systems [4] to improve speed of operation, decision-making transparency, and integrity control. Industry 4.0 aims to enhance the productivity of manufacturing management and service systems through employing technological advances in information technology (IT), such as artificial intelligence (AI) [5], to ensure that the right product is delivered at the desired “time, place, quantity, condition, and cost” [4].

A variety of mathematical programming algorithms have been utilized to tackle this challenge [1,6] and assist the warehouse operators in performing the task both effectively and efficiently. Unfortunately, the SLAP is a non-deterministic polynomial-time (NP)-hard problem where conventional optimization techniques require extensive computational time to find an optimal solution when the order size increases. Most optimization techniques applied to SLAP also derive a solution using a single *utility function* such as order-picking efficiency. These optimization techniques often neglect practical issues like rack stability [7] and human factor considerations such as worker safety [8].

The importance of incorporating human-centric solutions and safer working environments is a critical aspect of emerging strategy known as Industry 5.0 [9]. In the context of this industrial application, the allocation of storage spaces can significantly impact the safety of warehouse workers [10,11], particularly concerning hazards such as *falling objects*. According to the U.S. Bureau of Labor Statistics, the injury rate among full-time workers in the warehousing and storage sector was 4.8% in 2020. Furthermore, the *Occupational Safety and Health Administration* (OSHA) of the United States reported that over 50,000 individuals sustain injuries from falling objects in warehouses every year [12]. While various preventive measures have been implemented to enhance worker safety, accidents predominantly occur due to human errors. For instance, allocating a bulky or heavy item to an elevated rack increases the risk of injury. AI-based technologies represent an opportunity to mitigate safety concerns in warehouse environments [13] by providing safety-oriented optimal solutions to the industrial SLAP.

In industrial warehouses, different storage strategies are used when assigning incoming products to locations in the shelving units. These include *dedicated*, *flexible*, and *class-based* placement. Dedicated placement occurs when the identified products are always stored in fixed locations whereas flexible placement refers to products whose locations on the shelving units can vary over the time. Alternatively, class-based placement involves

allocating each item to a predetermined zone where its location in the identified zone can be flexible. Several policies have been introduced to allocate incoming items to available locations. These policies tend to rank items based on one or more criteria and prioritize the products to be placed near the *pick-up/drop-off* (*P/D*) for efficient retrieval. In addition, the policies often attempt to locate similar types of items close to each other on the storage shelves. Other approaches incorporate a hybrid criterion for rack assignment, such as the *cube per order index* (COI) [14,15], which considers both product turnover and volume.

The objective of this research is to provide an efficient AI-based (i.e. meta-heuristics) approach to a real-world storage allocation problem. The problem is modeled using mixed integer nonlinear programming with four utility functions including worker safety. The inclusion of worker safety as a utility function is critical and a novel aspect of this work. To achieve a meaningful result in practical time, a hybrid optimizer is developed, combining the benefits of *basin-hopping* and *simulated annealing* techniques. The hybrid optimizer is capable of handling both linear and nonlinear objective functions. The performance of the proposed technique is then compared with a well-established solver available to industry, i.e. SCIP, that can only utilize linear objective functions. The proposed approach provides warehouse managers with a flexible practical tool to store items based on their prioritized objectives. Section 2 summarizes the related published literature and discusses the need for optimized solutions to SLAP. The proposed methodology is explained in Section 3. The developed method is then applied to a real-world storage assignment problem found at an Australian furniture company and the outcomes are illustrated in Section 4. Section 5 discusses some key implications of the study, and section 6 presents conclusions and future work.

## 2. Related work

Industrial storage facilities and warehouses have often lacked appropriate planning and, therefore, the process of deciding the storage location for incoming pallets and products is not optimal [16]. To improve operational efficiencies, numerous medium and large storage warehouses have sought to optimize the process and reduce costs. Specifically, a great deal of interest has been directed toward optimizing the SLAP. Unfortunately, SLAP is an NP-hard problem due to various reasons including computational complexity [3]. Different policies have been introduced to facilitate the assignment decisions and consequently order-picking operation. Table 1 summarizes the six most common assignment policies used to assign the pallets of incoming orders to available (free) locations in the flexible storage facility [2,17].

**Table 1.** Common policies for assigning pallets to warehouse shelves [2,17].

Policy	Description
Random based	Each item is assigned to a free location across a warehouse with equal probability.
Closest available location	An item (of incoming order) is assigned to closest free space to the <i>P/D</i> point.
Affinity based	Items ordered mostly together by customers are located closer to each other.
Turnover or popularity based	Items with higher demand or popularity are placed closer to the <i>P/D</i> point.
Volume based	Items with larger volume are located closer to the <i>P/D</i> point.
Rule based	Items are stored based on rules extracted from human experience.

Most SLAP studies have investigated *order-picking efficiency* by minimizing the travel distance or traveling time of the item picker. The problem has been addressed using mathematical programming (e.g. integer programming, mixed integer linear, or non-linear programming), heuristic and meta-heuristic algorithms (e.g. greedy search and genetic algorithm (GA)), multi-criteria decision-making (e.g. ELECTRE and SMART), or a combination of these methods [1]. *Discrete event simulation* (DES) has also been used to evaluate the effects of storage assignment policies on warehouse performance [18]. DES has been utilized to simulate the performance of order-picking operations under various storage assignment policies, considering demand probability distributions [19], where the system was modeled using binary integer programming.

In some studies, an integrated approach that combined mathematical and heuristic optimization methods was employed. For instance, integer linear programming with a Lagrangian relax and fix approach was used to minimize warehouse costs [20]. The combination of *Non-Dominated Sorting GA* and *Ant Colony Optimization* was also developed for SLAP optimization problems, and results confirmed that this method outperforms common storage allocation policies such as random, closest open location, and ranked-based assignment [21]. In another study, integer and dynamic programming had been used for modeling SLAP, and the optimal solution was investigated by the *tabu search* algorithm [22]. The study showed that tabu search achieved better results than either a GA or a *greedy strategy*. Another method incorporated mathematical linear formulation and a heuristic local search strategy to find an appropriate solution for the large-scaled SLAP with compatibility constraints such as complementarity or affinity [23]. This approach was an extension of heuristic algorithms for warehouse storage with operational constraints [24].

The combination of SLAP with order-picking optimization and routing policies has also been modeled as cubic mixed integer programming problem and solved using a *General Variable Neighborhood Search* metaheuristic algorithm [25]. The study showed that the optimal storage allocation can reduce the picker's traveling time by 27% to 62%. In addition, an improved version of particle swarm optimization (PSO) had been proposed to solve joint SLAP, batching and picking

routing problem optimization [26]. The proposed method could obtain a solution in a short amount of time while improving traveling distance. Integer programming was also used to allocate SKUs to storage racks in earthquake-prone locations while minimizing order-picking travel distance and preventing racks collapse in seismic events [27].

In addition to order-picking, some researchers incorporated both picking comfort and picking cycle time in their model of the assignment problem [28]. In this work, the problem was modeled using integer programming and solved using a bi-objective optimization technique. The study showed that minimizing discomfort leads to an increase in cycle time. However, the merits of improving picking comfort and its effects on cycle time need a long-term study. Warehouse size determination has also been included in the assignment problem for simultaneous optimization. *Mixed integer linear programming* (MILP) was used for modeling the process while a heuristic approach was incorporated for optimization [29]. The proposed integration led to more efficient warehouse performance for severely constrained and large-scale problems [29]. A subsequent study focused on a new method of finding the optimum dimensions of a storage facility [30]. The researchers proved that a storage location policy did not affect warehouse shape optimization and could be studied independently. MILP was also utilized in a different SLAP study where the warehouse was served by milk-run logistics [31]. The objectives of the study were to minimize order cycle time and the pickers' time simultaneously. The technique had the ability to solve the problem using commercial optimization software, and achieved better results than a simple COI storage policy.

Few studies have proposed models with dynamic conditions which consider changeable demand patterns. These studies showed that optimization methods using meta-heuristic algorithms such as tabu search, simulated annealing, and greedy local search would lead to better warehouse performance under changing demand functions over time [2,32]. Affinity between items also adds complexity to the dynamic SLAP. A cluster-based assignment method had been developed to reduce retrieval time based on both turnover and affinity of items [33]. To address affinity considerations, *quadratic assignment problem* (QAP) approaches have also been used to model

dynamic SLAP with single objective, to minimize travel distance in a class-based storage scenario [2]. Due to computational complexity of the problem, the model solved by meta-heuristic algorithms (i.e. greedy GA) [34] where the affinity of items, called *association index* (AIX) [35,36], were extracted and the results showed significant improvement in order-picking travel distance and efficiency of the meta-heuristic algorithm in solving this complex problem.

Some warehouses incorporate zones of products which make optimization challenging using the previously discussed approaches. For example, consider dividing a warehouse into two-class storage space with reserve and forward areas, where the forward area is smaller than the reserve section. Products are stored in the reserve section, and items are frequently picked from the forward area to be replenished by inventories kept in the reserve section, hence making it challenging to allocate SKUs in both zones. One approach to tackle this problem is modeling the warehouse as *Knapsack problem* with the objective of minimizing the replenishment and picking costs [37]. In another class-based storage optimization approach, a new assignment policy used a cascade structure for allocating non-commodity items with recurrent but uncertain demands [38]. The study relied on the probability of revisiting a mold in a same location obtained from historical data and used mathematical programming for optimization. Additionally, warehouses can be divided into different zones based on familiarity between products in class-based optimization. To classify products, three criteria have been used to measure familiarity: *complementarity*, *compatibility*, and *popularity* [23]. Complementarity considers that whether items are commonly ordered together, such as a dining table and its chairs in a furniture warehouse. Compatible items possess at least one common feature, warranting their placement in proximity to each other – consider refrigerated and non-refrigerated products. Popularity, on the other hand, evaluates how frequently products are retrieved from their respective locations. This criterion enables the assignment of fast-moving or slow-moving items to a designated zone.

Collecting input data for SLAP optimization especially in large-size warehouses is time-consuming and must be performed precisely to be reliable. *Radio frequency identification* (RFID) as a means of data collection is proposed to facilitate capturing of real-time information in a warehouse [39]. An integrated system of gathering real-time data using RFID technology and PSO for SLAP shows the positive effects of real-time data gathering on the optimization of the problem with the objectives of cargo adjacency, item priority, and rack stability [40]. Some recent studies proposed novel approaches such as employing machine-learning techniques [41] and finite-

element-based method [42] for solving dynamic SLAP; however, their efficiency has not yet been tested on a real-world problem.

Potential health and injury risks are increasing due to the rapid growth and labor-intensive nature of warehousing operations; however, existing research on warehouse safety is scarce [13]. There is a significant negative relation between the presence of safe work procedures (as one of the safety management practices) and safety performance outcomes, such as accident rates [43]. In addition to costing the company more time and money, the appropriate safety measures will likely reduce the performance of the warehouse operations [44]. A framework for improving warehouse safety was introduced based on the risks generated in warehouse operations [45]. Monitoring the technical parameters of racks integrated with analytical simulation was proposed as a method of securing safety in high-level warehouses [46]. Notably, a study demonstrated that the number of accidents in warehouses was reduced by analyzing images captured by 360-degree cameras mounted on forklifts using a deep-learning algorithm [47]. To further enhance worker safety in cold-storage environments, deep neural network technology was combined with digital twin technology using data received from *Internet of Things* sensors [11].

Rarely did these studies tackle the challenge of including the human and safety in the warehousing optimization process. Worker safety is an inevitable consideration that must be more respected in warehouses with manual storing/retrieval processes; however, safety has not been considered in existing SLAP optimization approaches. The key contributions of this paper are as follows: introducing four utility functions for measuring and improving the defined objectives in SLAPs; developing an AI-based algorithm for discrete optimization of high-dimensional SLAPs with non-linear functions; and incorporating worker safety (as a systematic consideration) to the SLAP of a warehouse with a manual picker-to-part order-picking method.

### 3. Methodology

The proposed methodology aims to solve the SLAP for large orders in a multi-level manual picker-to-part warehouse in reasonable time. Each order contains several palletized products which must be placed at different rack levels with the objective of maximizing order-picking efficiency while considering affinity and safety constraints. The proposed methodology includes both problem modeling and optimization steps. The SLAP optimization model is divided into three categories based on the availability of information prior to receiving orders [48]. These categories include (i) *SLAP based on Product Information* (SLAP/



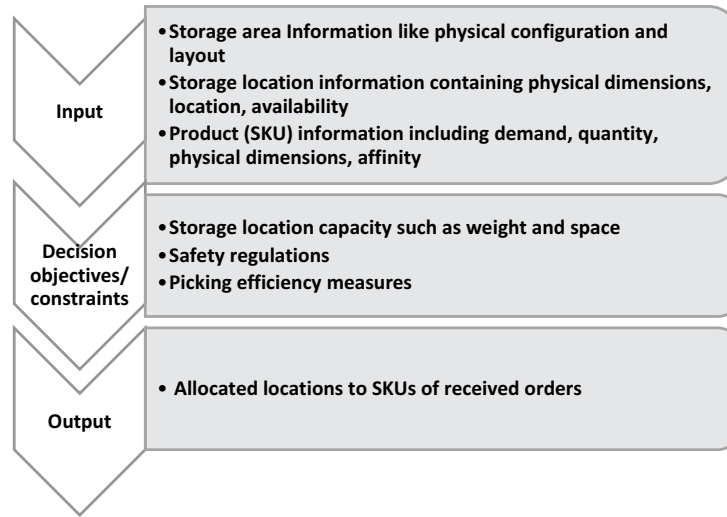


Figure 1. SLAP/PI decision model based on work by Gu et al. [48].

PI), when information at product level such as average number of picks or average duration of stay, is available; (ii) *SLAP based on Item Information* (SLAP/II), when item information such as time of arrival and shipping are known; and (iii) *SLAP based on NO Information* (SLAP/NI), when there is no information available for product allocation. Since product information is mostly available prior to receiving orders, the problem is modeled as SLAP/PI where three major components of the decision-making process are illustrated in Figure 1 [48].

Four objectives with respective constraints are extracted and modeled using a mixed-integer programming approach to achieve the appropriate allocation of pallets to empty rack locations. The objectives of *picking efficiency* and *affinity* facilitate the order-

picking operation by reducing the traveling distance and worker confusion about where to place the pallet. The objectives of *rack stability* and *worker safety* are also introduced in the proposed decision model to help ensure a safer work environment and, thereby, aligning the allocation decisions with the principles of Industry 5.0 [9]. The 27 notations used in describing the optimization process developed in this research are listed in Table 2.

### 3.1. SLAP optimization model

To develop the optimization model, the variables are normalized due to their different operating ranges. Linear (max) method is used for normalization where

Table 2. Notations used in this paper (Iv: integer variable, BI: binary integer variable, Rv: real variable, Sv: set or an array of variable).

Notation	Type	Description
$SL_j$	Sv	Storage location includes $\{x_j, y_j, z_j, Av_j, MW_j, d_j\}$
$x_j, y_j$	Rv	Horizontal coordinates of location $j$ in a warehouse
$z_j$	Rv	Vertical coordinate of location $j$ in a warehouse
$Av_j$	BI	Status of location $j$ that shows either the location is empty or occupied
$CW_j$	Rv	Maximum load capacity of location $j$
$d_j$	Rv	Distance between location $j$ and $P/D$ point
$PL_i$	Sv	Pallet containing product (SKU) $i$ which includes $\{w_i, D_i, r_i, q_i\}$
$w_i$	Rv	The weight of $PL_i$
$D_i$	Rv	Annual demand of SKU (product) $i$
$r_i$	Iv	Falling risk of SKU (product) in $PL_i$
$q_i$	Iv	Quantity of SKU in $PL_i$
$k_{i,j}$	BI	Occupancy variable that shows if $PL_i$ is assigned to location $j$ or not
$SO_i$	Sv	Solution in $i^{th}$ iteration of optimization or a set of occupancy variables $(k_{i,j})$ that meets problem constraints
$Q_i$	Rv	Relative fluctuation of utility values in $b$ consecutive solutions before $i^{th}$ iteration
$E$	Sv	Set of empty locations in the warehouse
$E_m$	Sv	Set of empty locations in $m^{th}$ the warehouse
$I$	Sv	Set of available PLs in a received order
$I_s$	Sv	Set of available SKUs with affinity in a received order
$f_{i,r}$	Rv	Penalty (cost) of not placing SKUs $i$ and $r$ in a same rack
$N$	Iv	Number of pallets in the incoming order
$T_i$	Rv	Optimizer temperature in $i^{th}$ iteration of optimization
$\alpha$	Rv	Geometric cooling rate
$\beta$	Rv	Vertical distance adjustment factor
$\hat{p}$	Rv	Relocation probability
$\delta$	Iv	Temperature update factor (after perturbations)
$n$	Iv	Maximum number of perturbations in a run of optimization

$v'$  shows the normalized value of variable  $v$ . To include *rack stability* and *safety* optimization in the proposed SLAP model, two utility functions (in Cobb-Douglas format) are defined incorporating warehouse safety regulations [49,50] and the objective function is defined by Eqn. (1). The constraints are given by Eqns. (2) to (5).

$$\text{Minimize : } W = \frac{1}{N} \sum_{i \in I} \sum_{j \in E} (k_{ij} \times w'_i \times z'_j) \quad (1)$$

Subject to the following constraints :

$$\sum_{j \in E} k_{ij} = 1 \forall i \in I \quad (2)$$

$$\sum_{i \in I} k_{ij} \leq 1 \forall j \in E \quad (3)$$

$$w_j k_{ij} \leq CW_j \forall j \in E \quad (4)$$

$$k_{ij} \in 0, 1 \forall i \in I, \forall j \in E \quad (5)$$

In this approach, maximizing rack stability is achieved by minimizing the *rack instability utility function* ( $W$ ). The function prioritizes the assignment of heavier pallets to lower rack locations, which also reduces the rack center of gravity. Furthermore, high rack stability reduces the probability of rack collapsing in an accident or during the occurrence of natural disasters.

Note that the utility function for measuring instability ( $W$ ) is inspired from a formula for calculating gravitational energy. The constraint given by Eqn. (5) defines the bounds and limits the occupancy variable so that the result is restricted to only binary values, while the constraints associated with Eqns. (2) and (3) ensure that only one unit load is assigned to each free location and all incoming pallets find a location in the warehouse. An allocated pallet must not weigh more than the rack location weight capacity which is represented by the constraint defined by Eqn. (4). In addition to stability, extracting heavy items from high rack levels often consume more time or energy which is not examined in this study.

The *worker safety utility function* ( $R$ ) is defined by measuring the risk of falling objects which threatens workers health while retrieving or storing an item. This risk is higher in products with uneven weight distribution, having bulky non-cuboid boxes, or with unsafe contents, such as dangerous chemicals. These hazards can pose risks to pickers handling products as well as to other

workers within the warehouse. For instance, in a furniture warehouse, a falling sharp-edged item from a top rack level can escalate the severity of potential injuries. This risk can be reduced by allocating high risk products to lower rack levels. For this, each product is allocated a falling risk number between 1 and 9 by an expert such as warehouse operation managers, where 1 represents items with lowest risk and 9 shows items with highest falling hazard shown in Table 3.

The objective function used to minimize  $R$  while observing constraints in this step is given by,

$$\text{Minimize : } R = \frac{1}{N} \sum_{i \in I} \sum_{j \in E} (k_{ij} \times r'_i \times z'_j) \quad (6)$$

Subject to : (2), (3), (4), (5)

To improve order-picking efficiency, items with high demand and small size must be placed closer to the  $P/D$  point from where the orders are shipped. In this way, the travel distance of retrieving items and, consequently, the total order cycle time is decreased [14]. Using high-level racks causes some difficulties in modeling the problem like considering order-pickers different vertical and horizontal speed (7). The distance of location  $j$  from  $P/D$  point ( $O$ ) in a three-dimensional warehouse is calculated by the introduced revised Manhattan method given by

$$d_j = |x_j - x_o| + |y_j - y_o| + \beta |z_j - z_o| \quad (7)$$

Since the horizontal and vertical speed of transporters is not equal in a pick-up trip,  $\beta$  as an adjustment factor is introduced for maintaining the equation consistency. For instance, if the average horizontal speed of an "order picker" (a transporter driven by the human used for picking several items in one trip) is 3 m/s and vertical speed is 0.5m/s,  $\beta$  can be set to  $3/0.5 = 6$ .

*Order-picking efficiency utility function* ( $D$ ) is used to measure the closeness of desirable items to the  $P/D$  point. Preferred items are smaller and have high frequent retrievals in a warehouse. The objective function for minimizing the total travel distance for picking orders is given by

$$\text{Minimize : } D = \frac{1}{N} \sum_{i \in I} \sum_{j \in E} (k_{ij} \times D'_i \times d'_j \times q'_i) \quad (8)$$

Subject to : (2), (3), (4), (5)

Order-picking typically accounts for 55% of warehouse operating costs [51] and its optimization is highly pursued in warehouse operations.

**Table 3.** Risk number scale.

Risk no.	1-2	3-4	5-6	7-8	9
Severity	Minor	Low	Moderate	High	Very High

In addition to above-mentioned objectives, some SKUs are preferred to be stored in a same rack or an aisle which are called *affinitive SKUs*. For instance, in the furniture industry, metal legs of a table and its top mounting glass plate may be received in different pallets but are preferred to be placed in a same rack to facilitate order-picking process. To include this preference, a non-linear utility function is defined as a penalty (cost) function to penalize the *affinitive SKUs* which are not allocated to a same rack. The objective function, which minimizes the normalized sum of associated penalties, called *affinity optimization*, is given by

$$\begin{aligned} \text{Minimize : } P & \\ &= \frac{\sum_{i \in I_s} \sum_{r \in I_s} \sum_{j \in E_m} \sum_{g \in E_m} (k_{ij} \times (1 - k_{r,g}) \times f_{i,r})}{\sum_i \sum_r (f_{i,r})} \forall s \\ &= s, \forall m \neq m \end{aligned} \quad (9)$$

Subject to : (2), (3), (4), (5)

### 3.2. Optimization

Solving this NP-hard problem exactly in large warehouses is not feasible; therefore, AI-based optimization approaches utilizing meta-heuristic algorithms are more applicable. This study developed a method inspired by *basin-hopping* (BH) [52] and *simulated annealing* (SA) [53] to find a desirable solution in a reasonable time. In the proposed algorithm, local search is performed using an SA concept while the random jump (perturbation) is inspired by BH algorithm. Figure 2 demonstrates how the proposed algorithm can investigate the search space for the global optimum solution.

The first step is setting up algorithm hyperparameters and experimental conditions such as optimizer initial temperature, stop criteria, cooling schedule and cooling rate, and after that the algorithm starts with generating a feasible initial solution which met all model constraints. The optimizer finds a random neighbor for the solution ( $SO_i$ ) considering the specific problem constraints. The neighbor solution is produced by one of the following random mechanisms: relocating a placed pallet to an empty location or swapping locations of two allocated pallets. If the neighbor has a better utility value ( $UV$ ), the solution is kept as the best solution, otherwise (if the neighbor has worse  $UV$ ), the neighbor is kept as the best solution with the probability of  $e^{\Delta UV/T_i}$  [54] where  $\Delta UV = (UV_{i-1} - UV_i)$ . The optimizer checks if the sum of relative fluctuations ( $Q$ ) of “ $b$ ” consecutive solutions, Eqn. (10), during the optimization of an objective function ( $F$ ) is smaller than a certain predetermined value ( $\epsilon$ ) and if the maximum number of perturbations is not reached. If so, the solution is perturbed, and optimizer

temperature will reset to a fraction of the initial temperature given by Eqn. (12).

If  $Q \geq \epsilon$ , the optimizer updates  $T_i$  and continues iterations, otherwise the algorithm stops iterations and returns the optima. If  $Q < \epsilon$  and the maximum number of perturbations is reached, the algorithm continues with iterations until the optimizer temperature is less than final temperature. The relative fluctuation of the utility values in  $b$  consecutive solutions before the  $i^{\text{th}}$  iteration is given by,

$$Q_i = \sum_{j=0}^{b-1} \frac{|F(SO_{i-j-1}) - F(SO_{i-j})|}{F(SO_{i-j-1})} \quad (10)$$

The perturbation mechanism ( $Pt$ ) is defined as random swapping of the location of a certain percent of items ( $\hat{p}$ ) (also called *relocation probability*) considering problem constraints. Maximum number of perturbations ( $n$ ),  $\epsilon$ , and  $\hat{p}$  are the model hyperparameters defined by the user.

Geometric cooling, Eqn. (11), is a common type of cooling schedule where  $\alpha$  is normally between 0.8 and 1. The larger value of  $\alpha$  means the slower cooling schedule and the lower chance of getting trapped in a local optimum [54]; however, the optimization time would increase.

$$T_i = \alpha T_{i-1} \quad (11)$$

$$T_i = \frac{T_0}{\delta * \text{number of performed perturbations}} \quad (12)$$

The optimizer initial temperature ( $T_0$ ) can be determined by the maximum difference between utility values of two neighbors ( $\Delta UV_{max}$ ) [54]. The proposed SLAP optimization algorithm is shown in Algorithm 1. The initial setup is executed in the first 5 lines of the algorithm, while lines 6 to 13 illustrate the simulated annealing component of the optimizer. After appropriate number of iterations, the best solution achieved in line 12 is utilized as a seed structure for perturbation. The new structure is generated by random displacement of the seed structure [55] which is defined as perturbation mechanism. Lines 14 to 16 depict the perturbation part, inspired by basin hopping algorithm.

## 4. Evaluation of SLAP optimizer

### 4.1. Data gathering and design of experiments

The efficiency of the proposed algorithm is evaluated by solving the storage assignment problem in a flexible storage warehouse of an Australian furniture company. Data from this case study were used to generate realistic test cases for the experimental evaluation. Input data such as bays’ locations are extracted from warehouse layout drawings. The floor plan of the warehouse has 65 five-to-six level racks, and each rack contains 240 to 336



Algorithm 1. SLAP optimization algorithm inspired by Basin hopping and Simulated Annealing

```

1  Begin
2  Set model hyperparameters
3   $SO_i \leftarrow$  Initial feasible solution
4   $UV_i \leftarrow F(SO_i)$  #Utility function value for  $SO_i$ 
5   $B \leftarrow SO_i$  #Set the final solution
6  While termination condition is not met
7   $SO_{i+1} \leftarrow N(SO_i)$  #Find a feasible neighbor
8  if  $UV_{i+1}$  is better than  $UV_i$ 
9   $B \leftarrow SO_{i+1}$ 
10 else
11 if  $r(\text{random } 0-1) < (e^{\Delta UV/T_i})$  #Accept worse solution
12  $B \leftarrow SO_{i+1}$ 
13  $T_{i+1} \leftarrow G(T_i)$  #Update optimizer temperature
14 if  $Q_i < \varepsilon$  and perturbation is allowed
15  $SO_{i+1} \leftarrow \text{Pt}(SO_i)$  #Perturb the solution
16  $T_{i+1} \leftarrow T_{pi}$  # Reset temperature to defined value
17  $i \leftarrow i + 1$ 
18 end
19 return B
20 end

```

bays or storage locations ( $SL_j$ ) with different weight capacities, heights, and distances to the  $P/D$  point.

A placed order is received in one or more containers, where each container can hold up to 21 pallets and each pallet contains one or more sales units. Order (I) and storage location information is captured by the company's warehouse management system (WMS). A pallet must be placed in one empty location (where  $Av_j = 0$ ) and items in a pallet would be retrieved in sales units such as cartons. Four racks of the company warehouse are selected as potential locations in this study and the WMS shows 169 empty locations in selected racks. The warehouse has a picker-to-part manual retrieval system [56] where workers pick items from racks and take them to shipping area by forklifts or Order Pickers. Considering the transporters speed,  $\beta$  is set to 6 in this problem.

Eight experiments have been designed to evaluate the proposed approach considering the four objectives: picking efficiency, affinity, rack stability, and worker safety. Table 4 summarizes the eight experiments performed in this study. The experiments are designed such that the small and large incoming orders contain 15 and 90 pallets, respectively, varying in weights, safety risk, annual demand, and quantity per pallet that should be assigned by the proposed algorithm. An example of small-order specifications is shown in Table 5, and it is assumed that the large order contains six small orders, i.e. the algorithm must assign incoming pallets to 15 or 90 empty bays (out of 169) across the warehouse optimizing three introduced single objectives. It is assumed that at least one storage location can be found to accommodate each pallet in the defined experiments.

A grid search method is used to set the hyperparameters of the algorithm. The optimizer initial temperature is calculated using  $\Delta UV_{max}$  in the following optimization sections. The cooling schedule is selected to have a rate of  $\alpha = 0.998$  to ensure very slow cooling process. The stop criterion is determined by the algorithm target temperature ( $T_f$ ), i.e.  $T_f = 0.01 \times 10^{-3}$ . The value of  $T_f$  is close to zero but never reaches absolute zero because of the geometric cooling schedule used to define the optimizer temperature. The numbers of perturbations ( $n$ ),  $b$ ,  $\delta$ , and  $\hat{p}$  are obtained experimentally (using grid search method) in this problem, and they are set to 3, 100, 4, and 0.25, respectively, for all experiments. The pre-determined value for the experiments is  $\varepsilon = 0.01$  for Experiments 1 and 2,  $\varepsilon = 0.03$  for Experiments 3 to 6, and  $\varepsilon = 0.1$  for Experiments 7 and 8. The penalty for not placing affinitive items on the same rack is assumed a constant value and due to the random nature of this optimization, all experiments are conducted 10 times to examine the variation of results. The experiments are performed on a commercial computer using Intel® Core™ i5-1035G1 CPU with base clock speed of 1.00 GHz and maximum turbo frequency of 3.6 GHz and 8 GB of memory.

#### 4.2. Optimization for order-picking efficiency

The objective is to minimize the distance utility function or distance paved by transporters for picking the customers' orders. The optimizer initial temperature for the proposed optimization is set to 1 considering the maximum possible difference between values of the utility function ( $D$ ) for two neighbors ( $\Delta UV_{max}$ ) [54]. This maximum difference is determined by moving

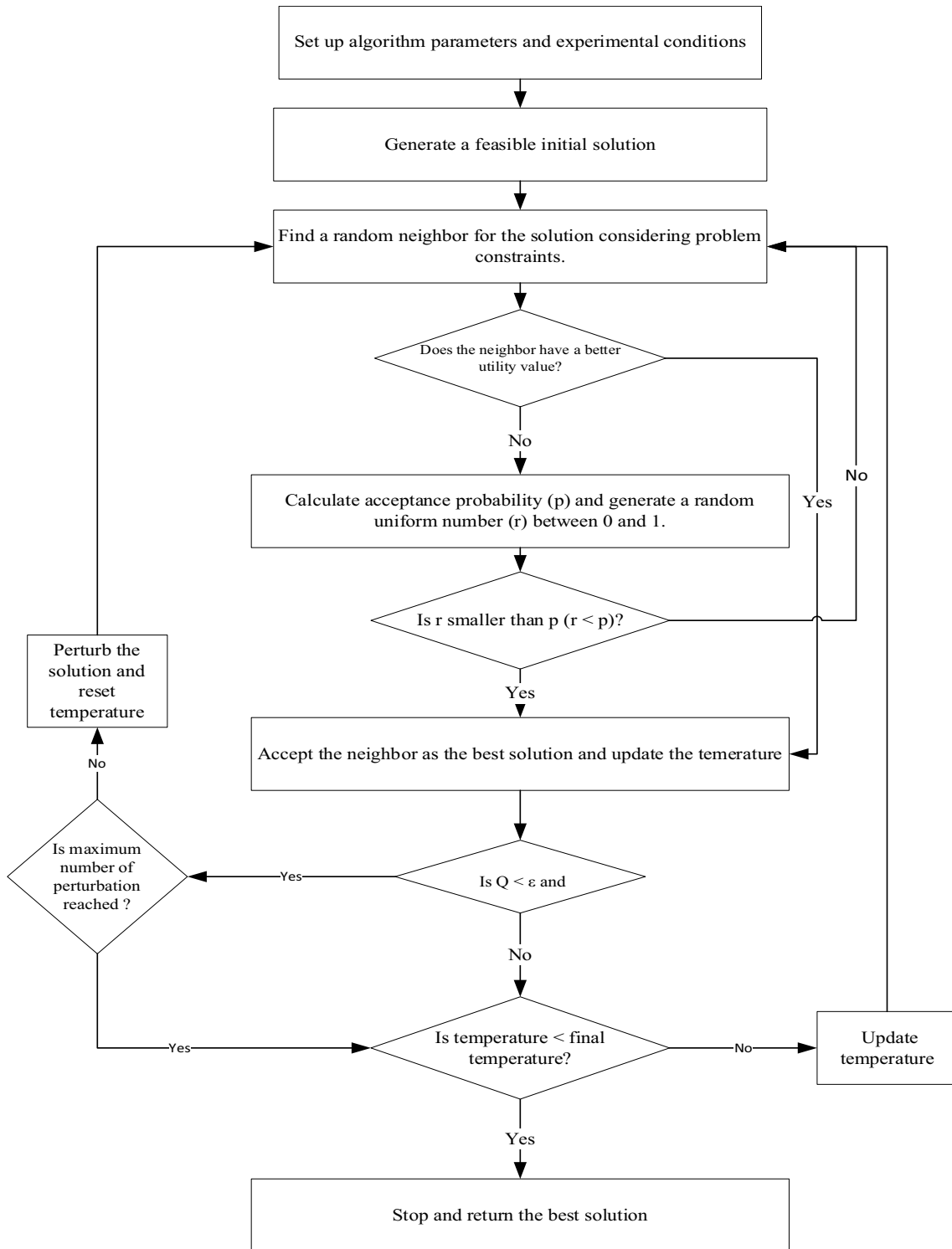


Figure 2. Flowchart showing the SLAP optimization algorithm.

Table 4. Experiment order, sizes, and objectives.

Experiment	Order Size (pallets)	Objective
1	15 pallets	Order picking efficiency
2	90 pallets	Order picking efficiency
3	15 pallets	Rack Stability
4	90 pallets	Rack Stability
5	15 pallets	Safety
6	90 pallets	Safety
7	15 pallets	Items affinity
8	90 pallets	Items affinity

Table 5. Example of small-order specifications.

SKU(i)	PL ID	PL <sub>i</sub>			
		$w_i(kg)$	$D_i$	$r_i$	$q_i$
1	999101	767	82	5	70
2	999102	541	37	9	20
3	999103	493	10	8	70
4	999104	421	40	5	50
5	999105	381	87	8	20
6	999106	365	32	9	100
7	999107	346	95	9	10
8	999108	342	70	9	80
9	999109	332	87	4	50
10	999110	326	95	5	30
11	999111	289	31	7	40
12	999112	265	34	3	60
13	999113	180	72	3	20
14	999114	58	74	3	70
15	999115	22	53	1	70

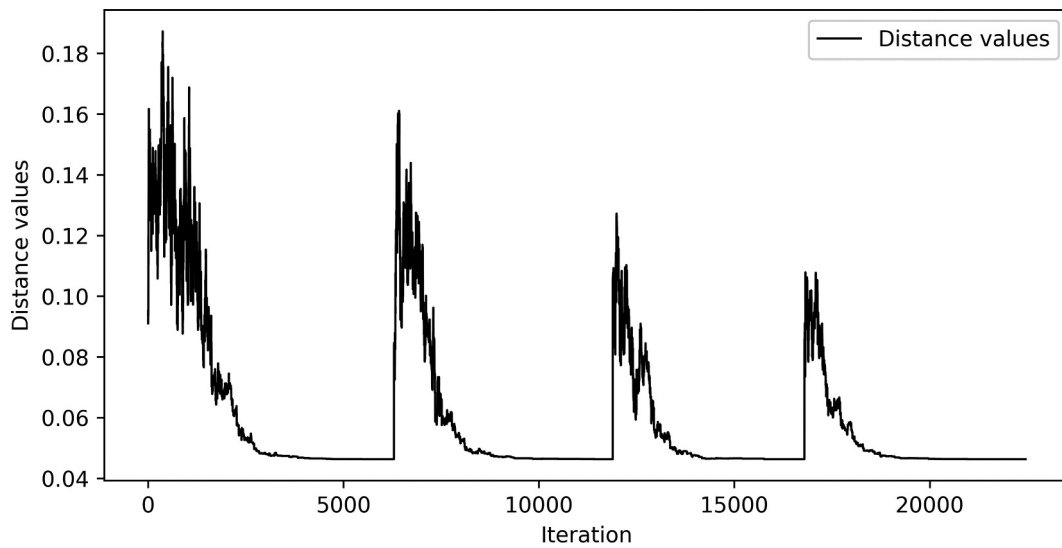


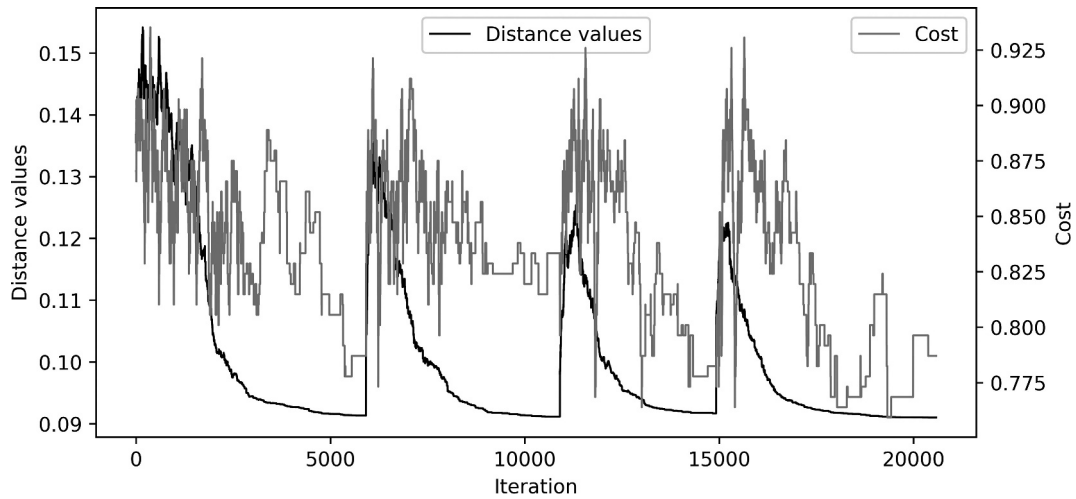
Figure 3. Distance ( $D$ ) optimization in Experiment 1: optimizing distance utility function for incoming order of 15 PLs varying in weight, safety risk, annual demand, and quantity per pallet. The algorithm manages to find a solution around 20,500th iteration.

a pallet with the highest demand score ( $D'_i \times q'_i$ ) from closest to the farthest possible empty location.

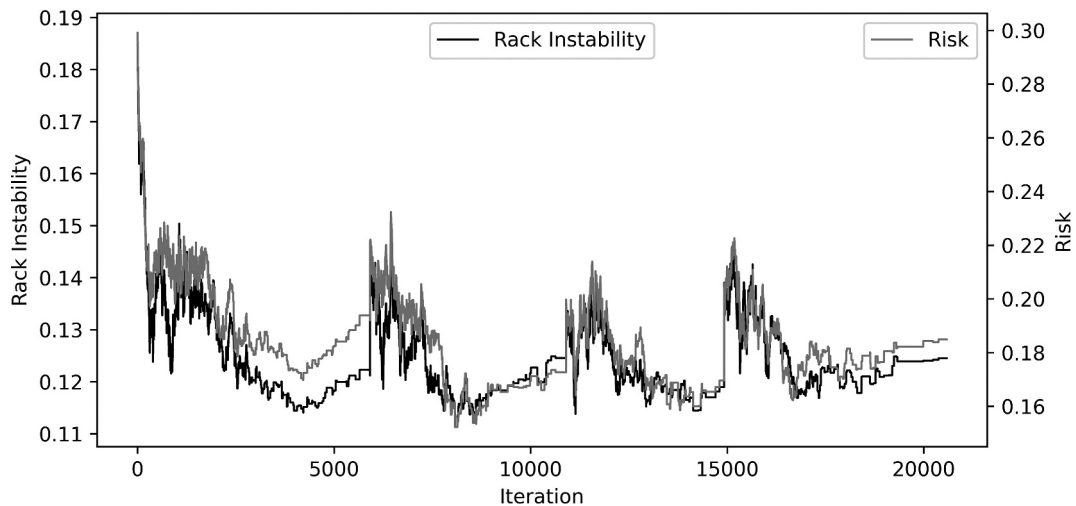
Experiment 1 is performed using a feasible initial solution where an incoming order must be placed in 15 out of 169 available locations with the objective of maximizing order-picking efficiency. The calculated initial value of utility function  $D$ , Eqn. (8), for this experiment is  $9.09 \times 10^{-2}$ . After repeating the experiment for 10 consecutive runs, the mean  $UV$  of final solutions is improved (i.e. decreased) by 49% to  $4.63 \times 10^{-2}$ . Standard deviation of final solutions is  $2.24 \times 10^{-5}$  which is less than 0.1% of the mean and the average execution time is 8 s. The slight variation of final value happens due to random nature of this optimizing process. As illustrated in Figure 3, the algorithm has hill-climbing capability and the tendency to accept worse neighbors during iterations with the probability of  $e^{\Delta UV/T_i}$  to prevent getting stuck in a local minimum.

In addition, three jumps, representing three perturbations, can be observed in the graph of distance utility values during the course of optimization. When the optimization is close to the last iteration in each perturbation, the probability of selecting worse solution decreases and less fluctuation is observed. Similarly, Experiment 2 runs 10 times using a feasible initial solution and the execution time takes 42 s in average. The mean  $UV$  of obtained solution decreased to less than 70% of its initial through iterations and the standard deviation of final solution is less than 0.2% of the mean. Figure 4 illustrates the fluctuation of distance and cost utility values in a sample run of Experiment 2.

In experiments 1 and 2, there is a notable fluctuation of  $Distance\ UV$  at the experiment's outset and following each perturbation, primarily due to the elevated optimizer temperature. Second and third perturbations experience less variation of  $UV$



**Figure 4.** Distance ( $D$ ) optimization for Experiment 2: allocating a large size order. The optimizer can escape from the local minimums and convergence happens after 18,500th iterations.



**Figure 5.** Rack instability ( $W$ ) and risk ( $R$ ) fluctuations during distance optimization for Experiment 2. Although a decrement is seen, it does not converge toward minimum.

due to having lower optimizer temperature at their beginning. Figure 4 also shows a possible correlation between distance and cost (affinity) optimization; however, it is not examined statistically in this study. An improvement of rack stability and safety is achieved during distance optimization due to the method of distance calculation. Since vertical distance is adjusted by  $\beta$  in the revised Manhattan method, the optimizer prefers to locate items in lower shelves which influences the solution to meet other problem objectives, Eqns. (1) and (6), partially. Figure 5 depicts the UVs of instability and risk through a run of the distance optimization in Experiment 2. The left vertical axis represents rack instability UV and the right one shows risk UV. The

perturbations of solutions are also evident as three distinct leaps in Figure 5.

### 4.3. Optimization of rack stability

This step aims at maximizing rack stability by minimizing the height of center of gravity considering allocation constraints. With distance optimization,  $\Delta UV_{max}$  for this utility function is calculated and initial optimizer temperature is set to 1. Figures 6 and 7 illustrate rack instability UVs in one run of optimizations and how the optimizer escapes from the trap of local optimums to satisfy the stability objective function. The optimizer found the global minimum of Experiment 3 in less than 7s where the utility value of minimum is 0.0 in all 10 runs. Zero

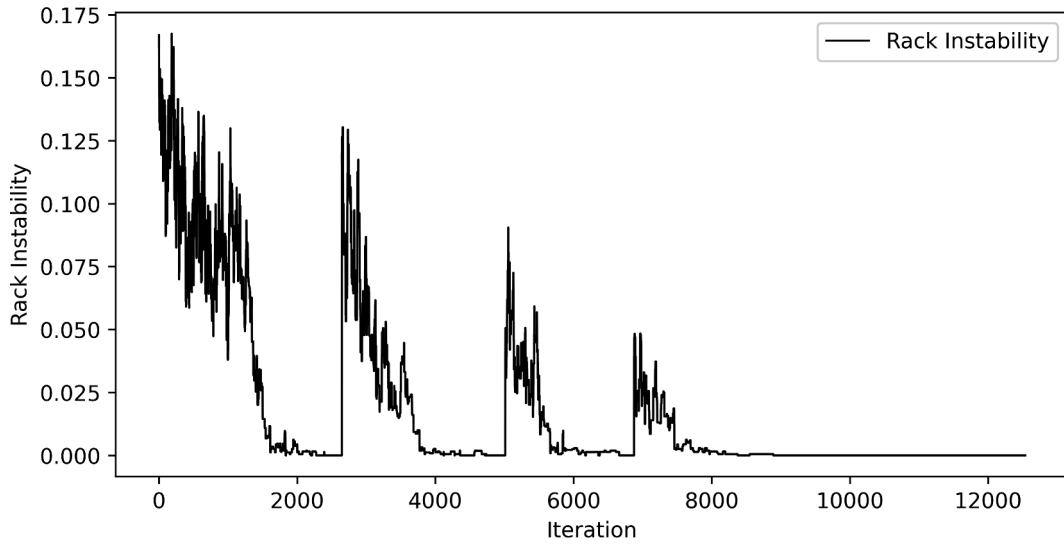


Figure 6. Rack instability ( $W$ ) optimization for Experiment 3. It converges toward the optima after 10,000th iterations.

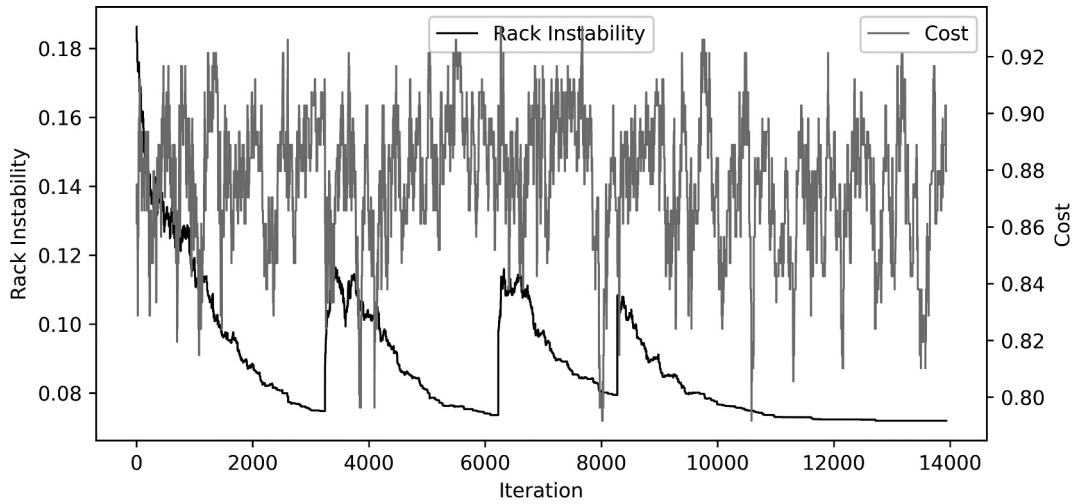


Figure 7. Rack instability ( $W$ ) optimization for Experiment 4. It converges toward the minimum after 13,000th iterations.

instability was attained in the optimization of the small order, as the optimizer managed to allocate all items on the ground level of the racks. In Experiment 4, the solution is achieved in 38 s, and final utility value of all runs is less than 38% of the initial value.

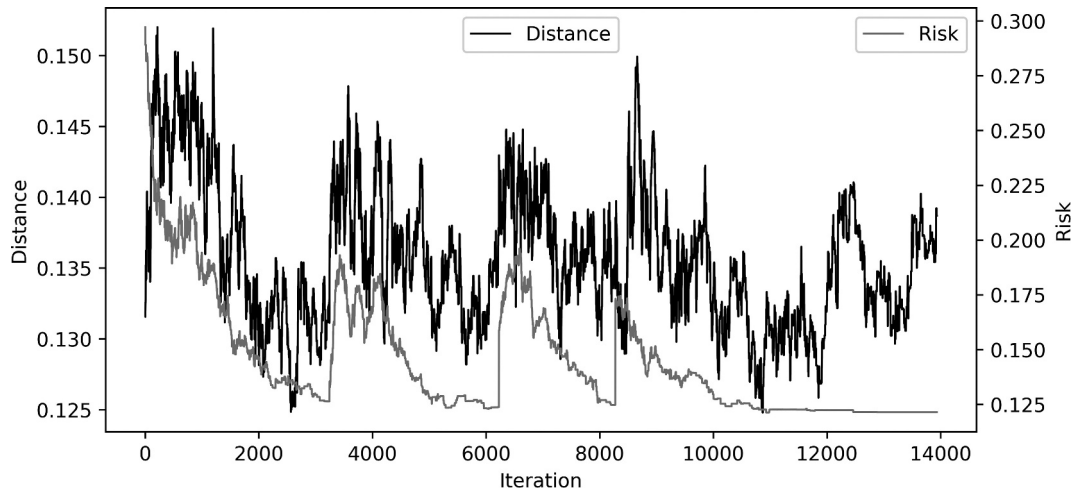
The standard deviation of final solutions is less than 0.03% of the average, which is acceptable for this problem. There is an observed improvement of storage risk during rack stability optimization as shown in Figure 8. This improvement must happen due to the existence of  $z_j$  in their objective function, Eqns. (1) and (6), and possible correlation between risk number and weight of an SKU, therefore locating heavy pallets in lower shelves can improve storage risk as well. There is no sign of correlation between stability and distance objectives, and only random fluctuation of distance UVs is observed in rack stability optimization. During stability optimization, cost (penalty) utility values

fluctuate randomly, and no specific pattern is observed according to Figure 7.

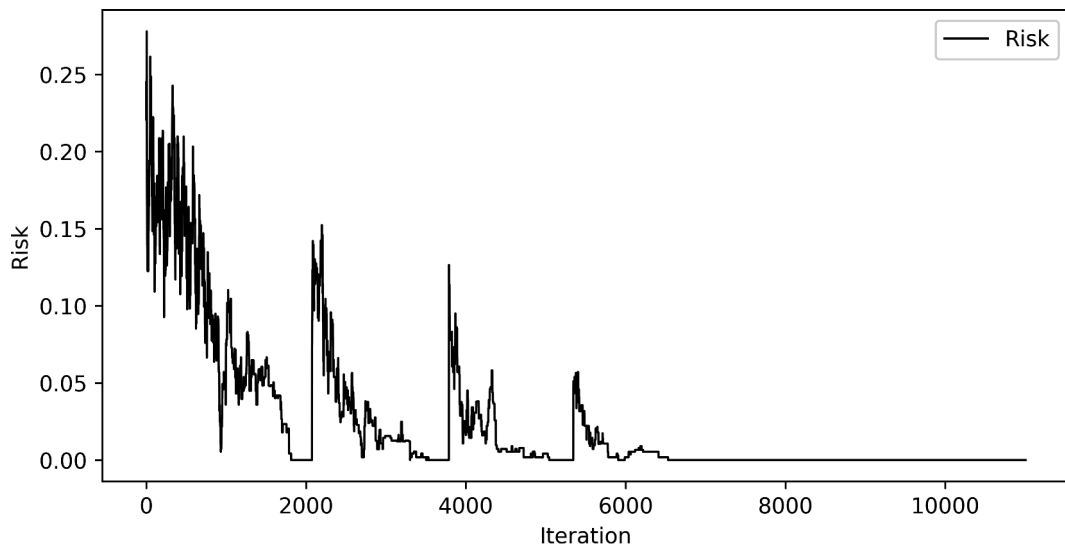
#### 4.4. Optimization for safety

To minimize the risk of falling objects in the warehouse, SKUs with higher risk numbers must be assigned to lower shelves. Optimizer initial temperature is adjusted to 1 which are calculated in the same way of optimizations in Sections 4.2 and 4.3. Experiments 5 and 6 are conducted for 10 times to examine the algorithm applicability and variation of the final solutions. Figure 9 depicts risk UVs during one run of Experiment 5. The utility value of a generated feasible initial solution reduces to the optimum value of 0.0 at the end of optimization in all runs. The average execution time is 8 s in the conducted experiment. Zero risk was achieved in the opti-

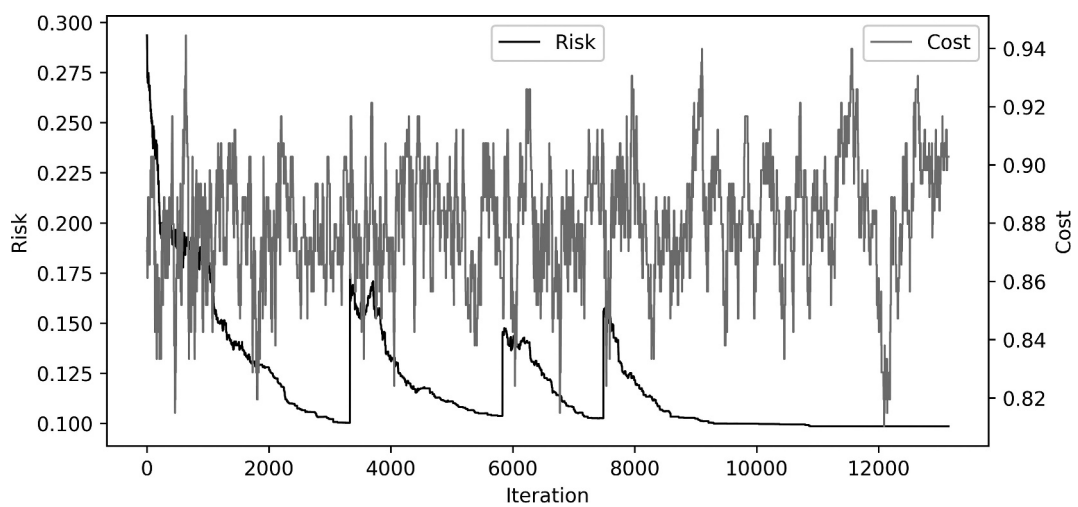




**Figure 8.** Distance ( $D$ ) and Risk ( $R$ ) values through stability optimization for Experiment 4. An improvement is observed for Risk ( $R$ ) but distance ( $D$ ) has random fluctuation.



**Figure 9.** Risk ( $R$ ) optimization for Experiment 5: wide range of fluctuation at the beginning is seen but it converges toward the final solution after 7,500th iterations.



**Figure 10.** Risk ( $R$ ) optimization for Experiment 6. It does not show a steep slope of improvement after 9,500th iterations and converges to the minimum after 11,500th iterations.

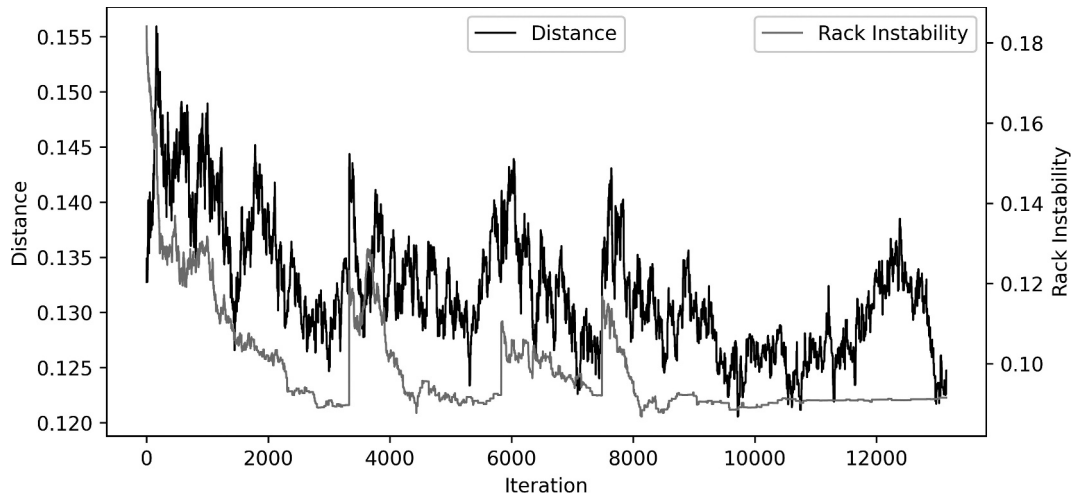


Figure 11. Distance ( $D$ ) and rack instability ( $W$ ) values through safety optimization for Experiment 6. Improvement is observed for instability, but distance depicts random variation.

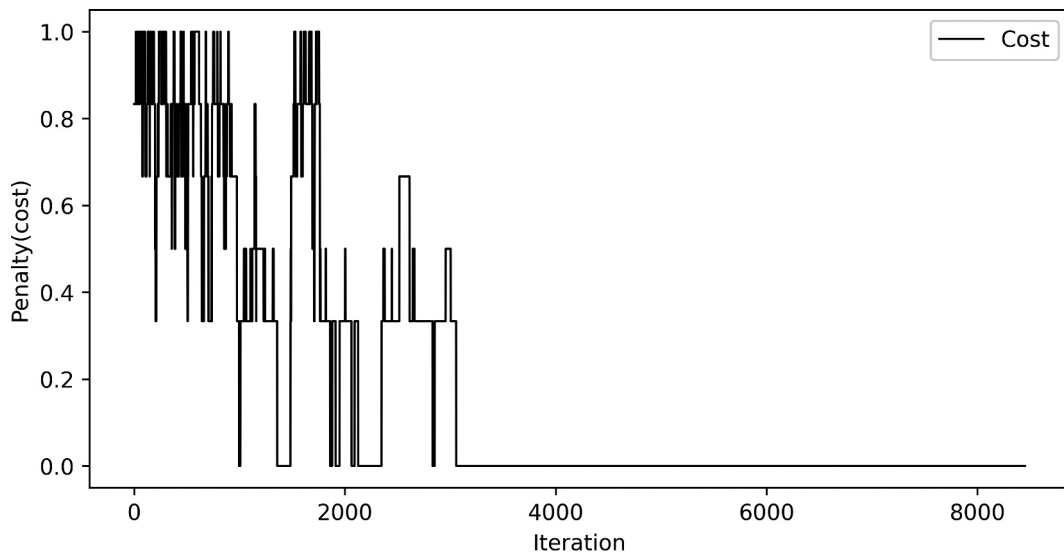


Figure 12. Penalty ( $P$ ) optimization for Experiment 7. The optimizer could place all affinitive items in a same rack.

mization of small orders, because the optimizer successfully allocated all items to the ground level of the racks, thereby completely eliminating the risk of falling objects.

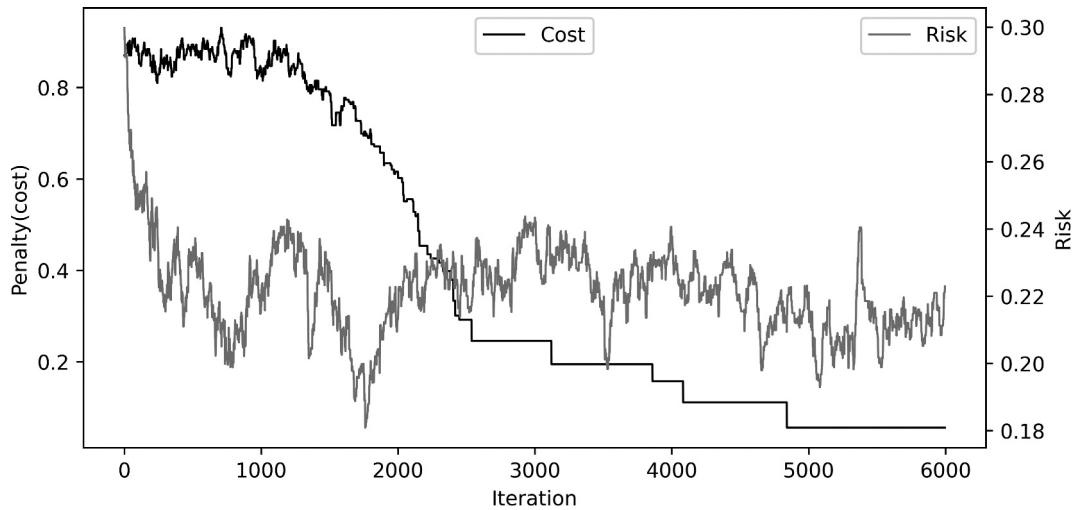
Comparing the safety utility value of the initial solution in Experiment 6 and the average UVs of 10 runs shows 49% decrement with the standard deviation of 0.04% in this optimization and average execution time was 34 s. Figure 10 demonstrates the improvement of safety UVs in a run of Experiment 6 while avoiding local minimum traps. There exists a random fluctuation in cost utility values, and no discernible correlation between safety and cost can be derived.

There is no evidence of an improvement of distance in the safety optimization, but rack stability

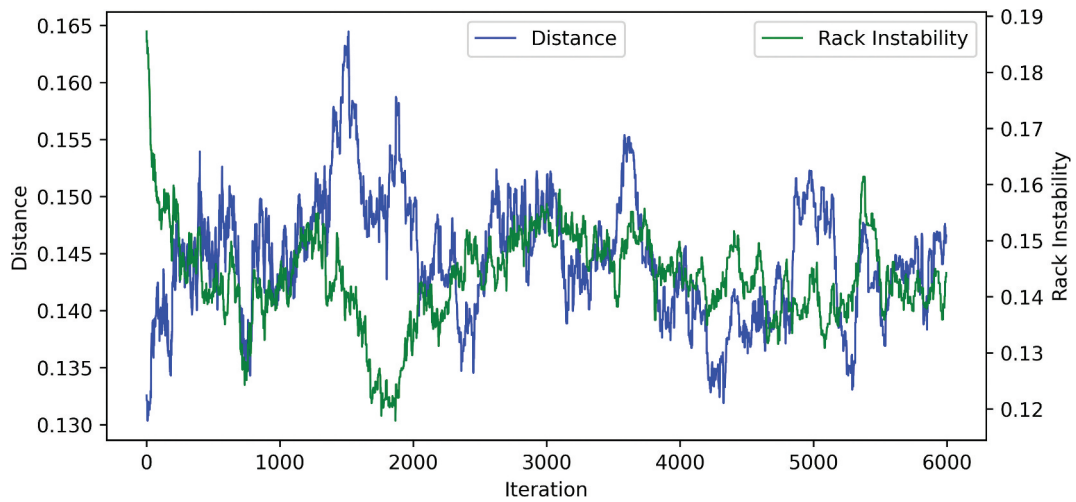
shows enhancement confirms existing correlation explained in Section 4.3. Figure 11 illustrates how distance fluctuates and stability improves in a run of Experiment 6.

#### 4.5. Optimization for affinity

It is preferable to assign affinitive or similar types of items in a same rack to avoid order-picking confusions and reducing order-picking time. Experiments 7 and 8 are conducted 10 times each in order to minimize the specified penalty (cost) function. Illustrative runs of the optimization for Experiments 7 and 8 are presented in Figures 12 and 13, respectively. For the small order (Experiment 7), the initial solution is 1 which means no



**Figure 13.** Penalty (P) and risk (R) optimization for Experiment 8. The optimizer could find the minimum penalty after 4,900th iterations.



**Figure 14.** Distance (D) and rack instability (W) UVs during affinity optimization. No specific correlations or patterns can be observed in the graph.

**Table 6.** Initial and final utility values of the problem objective in conducted experiments by proposed SLAP optimizer vs SCIP.

Experiment	Initial Utility Value	Achieved Utility Value (SLAP Optimizer)	Achieved Utility Value (SCIP)
1	$9.09 \times 10^{-2}$	$4.63 \times 10^{-2}$	$4.63 \times 10^{-2}$
2	$13.26 \times 10^{-2}$	$9.08 \times 10^{-2}$	$9.07 \times 10^{-2}$
3	$15.21 \times 10^{-2}$	0.00	0.00
4	$18.70 \times 10^{-2}$	$7.17 \times 10^{-2}$	$7.17 \times 10^{-2}$
5	$24.50 \times 10^{-2}$	0.00	0.00
6	$29.90 \times 10^{-2}$	$9.85 \times 10^{-2}$	$9.85 \times 10^{-2}$
7	1	0.00	N/A
8	$87.04 \times 10^{-2}$	0.00	N/A

affinitive items in a same rack. The optimizer could allocate all affinitive items in a same rack and find the minimum achievable penalty in this experiment which is zero.

The penalty utility value was  $87.04 \times 10^{-2}$  in initial allocation of experiment 8, and the penalty reduced to zero where the standard deviation of solutions is zero.

No huge leaps of penalty UVs are observed due to the larger value of  $\epsilon$  and the global minimum is found during minimization. Random fluctuations of rack instability and risk UVs are observed during penalty optimization as depicted in Figures 13 and 14. Although there might be possible correlation between penalty and distance optimization, the impacts of optimizing items affinity on order picking efficiency are not examined in this study and can be further assessed using discrete event simulation of order-picking process.

## 5. Discussion

While many existing studies on warehouse optimization yield valuable theoretical insights, they have often failed to benefit related industries due to their lack of consideration for practical factors. Storage allocation and safety measures were traditionally regarded as tactical decisions. The utility functions

**Table 7.** Mean values of the final utility functions for the SLAP optimizer after the completion of experiments.

Experiment	Distance	Rack Instability	Safety	Penalty (Cost)
1	$4.63 \times 10^{-2}$	$4.08 \times 10^{-2}$	$5.20 \times 10^{-2}$	$63.33 \times 10^{-2}$
2	$9.08 \times 10^{-2}$	$12.02 \times 10^{-2}$	$18.48 \times 10^{-2}$	$64.08 \times 10^{-2}$
3	$10.53 \times 10^{-2}$	0.00	0.00	$89.43 \times 10^{-2}$
4	$13.23 \times 10^{-2}$	$7.17 \times 10^{-2}$	$12.32 \times 10^{-2}$	$50.96 \times 10^{-2}$
5	$9.38 \times 10^{-2}$	0.00	0.00	$88.78 \times 10^{-2}$
6	$13.07 \times 10^{-2}$	$9.09 \times 10^{-2}$	$9.85 \times 10^{-2}$	$50.45 \times 10^{-2}$
7	$14.88 \times 10^{-2}$	$16.20 \times 10^{-2}$	$25.89 \times 10^{-2}$	0.00
8	$13.98 \times 10^{-2}$	$14.21 \times 10^{-2}$	$22.62 \times 10^{-2}$	0.00

**Table 8.** Execution time of examining total search space for small-order size.

Incoming Pallets	Search Space Size	Execution Time (Seconds)
1	169	<1
2	28392	<1
3	$4.74 \times 10^{-2}$	2
4	$7.87 \times 10^{-2}$	268
5	$1.30 \times 10^{-2}$	48,461

introduced in this research not only aid in measuring the quality of an assignment solution but also serve as objective functions for the developed optimizer to enhance allocation decisions. The proposed optimizer can solve the SLAP using linear or non-linear objective functions and outperforms the simulated annealing method. In fact, the optimization algorithm functions as a pure simulated annealing optimizer when the number of perturbations ( $n$ ) is set to zero. Nonetheless, zero (0) did not emerge as the optimal value for  $n$  during the hyperparameters tuning process. Furthermore, the optimizer possesses the capability to directly seek solutions for SLAP as long as the problem's objectives and constraints can be mathematically modeled. This eliminates the need to explore potential transformations from objective functions to constraints.

Table 6 provides the initial and the mean of best achieved utility values of the objective obtained by the proposed SLAP optimizer experiments described in Section 4. The same experiments are also conducted using one of the fastest well-known optimizers called *Solving Constraint Integer Programming* (SCIP) [57] adopted by Google OR-Tools [58]. The solver is open-source, noncommercial, and widely used for solving MILP problems.

Although SCIP achieved a slightly better solution in Experiment 2, the proposed algorithm was able to find comparable solutions to SCIP in all other experiments. Nevertheless, SCIP cannot find a solution for a SLAP with nonlinear objective functions (i.e. Experiments 7 and 8). In contrast, the proposed SLAP optimizer is able to find reasonable solutions. However, the execution time of finding a solution with SCIP is less than 6 s in all conducted experiments, which is faster than the SLAP optimizer developed in this research. Table 7

summarizes the mean of final utility function values after performing the defined experiments.

Industrial warehouses need to be able to adjust the problem objective or the storage location objective for different types of products over time. The results of this study highlight the flexibility and adaptability of the proposed methodology in addressing different warehouse optimization priorities. The impact of selecting different objective functions on the utility value ( $UV$ ) of the final solutions is illustrated in Table 7. For example, when the problem was optimized for safety (i.e. Experiment 6), the resulting risk utility value was 47% better than for the same problem where picking efficiency was optimized (i.e. Experiment 2). Conversely, the picking efficiency utility value for Experiment 2 was 31% lower (indicating better performance) than Experiment 6. The ability of decision-makers to examine the impact of different objective functions on the assignment problem will enable them to make informed decisions which can enhance picking efficiency or maximize safety when performing material handling operations in a warehouse. The proposed methodology will also enable decision-makers to allocate multiple incoming orders on an on-demand basis. In addition, SLAP optimization can be a competitive advantage in warehousing industries where efficient order fulfillment and safe work environment are critical.

In terms of computational time, investigating the total search space is the worst-case scenario often performed by optimization techniques like *Enumeration method of assignment* [59]. The search space size can be calculated by a permutation equation which is a function of two factorial expressions. Therefore, the complexity of the problem increases dramatically with the number of occupancy variables (i.e. the number of available locations and pallets). Table 8 shows execution time for allocating a smaller number of pallets to a storage facility with 169 empty locations by investigating the entire search space. It can be deduced from the results in the table that examining the entire search space for large-order

assignments, like a 90-pallet order, can quickly become impractical from a computational time perspective.

Although not as fast as the SCIP program, the proposed SLAP optimization method is able to find an industry acceptable solution for allocating a large order between 5 and 45 s. To achieve this time, the maximum number of solutions ( $m$ ) that are investigated with the proposed SLAP method using the geometric cooling schedule, Eqns. (11) and (12), can be calculated by

$$m = (n + 1) \times \ln\left(\frac{T_f}{T_0}\right) / \ln(\alpha) \quad (13)$$

which is less than 30,000 in all defined experiments in this study. Since the proposed optimizer does not examine all the search spaces, the execution time is still acceptable for real-world optimizations.

## 6. Conclusion

Artificial intelligence has been introduced as an essential tool of the fourth and fifth industry generations and improved efficiency of different sectors including the logistics industry. SLAP is a consistent challenge in medium to large size warehouses and proved as an NP-hard problem. In this study, the SLAP is modeled with mixed integer non-linear programming technique and is solved using a random optimizer inspired by basin-hopping and simulated annealing algorithm for discrete variable optimization. This paper focused on four practical objectives: order-picking efficiency, rack stability, worker safety, and affinity. The approaches presented in this paper optimize for each of these individually.

The proposed approach can find a reasonable solution for nonlinear problems and avoids local optima due to its hill-climbing feature. The optimizer can solve the SLAP and find desirable solutions in reasonable time, regardless of the equation type of objective functions or constraints. The experiments conducted in this study for allocation of small and large order size show that although distance optimization achieved a decent solution for the problem, only moderate improvement of safety and stability is observed during the optimization. This solution might not be acceptable for employers or safety regulators as they desire to protect workers safety in organizations. On the other hand, the proposed algorithm could obtain acceptable solution considering safety and rack stability objectives and an indication of positive correlation between minimizing storage risk and instability is observed through optimization. This correlation must be because of the possible dependency of risk number and SKU weight. But safety improvement did not have any impact on order-picking efficiency which is an enormous challenge for warehouse operation optimization. Furthermore, the performance of the proposed optimizer and SCIP solver has been compared, and it is

shown that the optimizer could achieve comparable solutions in most of the experiments; however, the SCIP could find final solutions more rapidly.

In future work, multi-objective SLAP optimization will be investigated to consider all objectives simultaneously. The multi-objective approach has already been studied in many similar fields such as resource allocation with *Quality of Service* constraints [60] or redundancy allocation problem [61] and has achieved appropriate results. Furthermore, the development of a mathematical or simulation model can be investigated to validate and measure the impact of employing the proposed approach on the order cycle time. Moreover, the proposed model is capable of incorporating different constraints and safety or non-safety objectives such as energy consumption to examine other aspects of the problem and expand it to more comprehensive optimization.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

Financial support for the project was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC grant RGPIN/04797-2022) and Goodwin Technology (H.K.) Co., Ltd.

## Notes on contributors

*Amir Zarinchang* is a PhD candidate at the University of Western Ontario, London, Canada. He holds a Bachelor's and a Master's degree in Industrial Engineering and Mechanical Engineering. His area of research is the optimization of operations and processes in manufacturing and logistics systems by utilizing artificial intelligence and machine-learning techniques.

*Kevin Lee* is a Senior Lecturer in Software Engineering and Internet of Things (IoT) at the School of Information Technology in Deakin University. He is the Course Director for the Bachelor of Software Engineering (Honours) and the Academic Director (teaching) for the School of IT. Previously, he held the positions at Nottingham Trent University, Murdoch University, the University of Manchester, and Lancaster University. He is a recognized researcher in the area of distributed systems and autonomous computing. Dr. Lee has worked in the areas of IoT, Cloud Computing, Robotics, Embedded Systems, cyber-physical systems.

*Iman Avazpour* is a Technical Lead (Data Operations) at Lexar. His area of research interest is Software Engineering and Data Transformations. He holds PhD in information communication technology, MSc in intelligent systems and robotics engineering, and Bachelor of software engineering.

*Dongxing Zhang* received the M.S. degree in aerospace manufacturing engineering from Beijing University of Aeronautics and Astronautics (China) in 2010 and a Ph.D. degree in Mechanical and Materials Engineering from the University of Western Ontario in 2017. Since 2021,



Dr. Zhang is an Associate Professor at Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC). His research interests include smart manufacturing, 3D printing, and precision manufacturing.

*Jun Yang* is currently a Professor at the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC) and Adjunct Professor in Mechanical and Materials Engineering at The University of Western Ontario, London, Canada. Dr. Jun Yang is a Fellow of the Canadian Academy of Engineering (FCAE). His research interests include additive manufacturing, micro/nanofabrication, and AI-based and internet-based intelligent manufacturing, metamaterials, and industrial Internet of Things.

*George K. Knopf* is a Professor in the Department of Mechanical and Materials Engineering at the University of Western Ontario, London, Canada. He has recently authored the book *Elastomeric Optics: Theory, Design, and Fabrication* (De Gruyter, 2022) and co-authored the books *Biofunctionalized Photoelectric Transducers for Sensing and Actuation* (SPIE Press, 2017) and *Light Driven Micromachines* (CRC Press, 2018). His research interests include bioelectronics, biosensors, degradable sensors, laser materials processing, process optimization, and flexible optical sheets.

## Data availability statement

The data that support the findings of this study are available upon a reasonable request from the corresponding author [GKK], with the permission of the industrial partner of this research. The data are not publicly available due to their confidential information of the industrial partner.

## Nomenclature

AI	Artificial Intelligence
AIX	Association Index
BH	Basin Hopping
COI	Cube per Order Index
GA	Genetic Algorithm
IoT	Internet of Things
IT	Information Technology
MILP	Mixed Integer Linear Programming
NP	Nondeterministic Polynomial time
OR	Operation Research
OSHA	Occupational Safety and Health Administration
PSO	Particle Swarm Optimization
QAP	Quadratic Assignment Problem
RFID	Radio Frequency Identification
SA	Simulated Annealing
SCIP	Solving Constraint Integer Programming
SKU	Stock Keeping Unit
SLAP	Storage Location Assignment Problem
UV	Utility Value
WMS	Warehouse Management System

## References

- [1] Reyes JJR, Solano-Charris EL, Montoya-Torres JR. The storage location assignment problem: a literature review. *Int J Ind Eng Comput.* 2019;10(2):199–224. doi: 10.5267/j.ijec.2018.8.001
- [2] Kofler M. Optimising the storage location assignment problem under dynamic conditions[dissertation]. Linz (Austria): Johannes Kepler University; 2014.
- [3] Frazelle EH. Stock location assignment and order picking productivity[dissertation]. Atlanta(GA), USA: Georgia Institute of Technology; 1998.
- [4] Barreto L, Amaral A, Pereira T. Industry 4.0 implications in logistics: an overview. *Procedia Manuf.* 2017;13:1245–1252. doi: 10.1016/j.promfg.2017.09.045
- [5] Mosallaeipour S, Nazerian R, Ghadirinejad M. Industrial Engineering in the industry 4.0 era. *Industrial Engineering in the industry 4.0 era.* 2018; pp. 63–81. doi: 10.1007/978-3-319-71225-3\_6
- [6] Rouwenhorst B, Reuter B, Stockrahm V, et al. Warehouse design and control: framework and literature review. *Eur J Oper Res.* 2000;122(3):515–533. doi: 10.1016/S0377-2217(99)00020-X
- [7] Vanheusden S, van Gils T, Ramaekers K, et al. Practical factors in order picking planning: state-of-the-art classification and review. *Int J Prod Res.* 2022;61(6):2032–2056. doi: 10.1080/00207543.2022.2053223
- [8] De Lombaert T, Braekers K, De Koster R, et al. In pursuit of humanised order picking planning: methodological review, literature classification and input from practice. *Int J Prod Res.* 2022;61(10):3300–3330. doi: 10.1080/00207543.2022.2079437
- [9] Ghobakhloo M, Iranmanesh M, Tseng ML, et al. Behind the definition of industry 5.0: a systematic review of technologies, principles, components, and values. *J Ind Prod Eng.* 2023;40(6):432–447. doi: 10.1080/21681015.2023.2216701
- [10] Aneziris ON, Papazoglou IA, Mud M, et al. Occupational risk quantification owing to falling objects. *Saf Sci.* 2014;69:57–70. doi: 10.1016/j.ssci.2014.02.017
- [11] Zhan X, Wu W, Shen L, et al. Industrial internet of things and unsupervised deep learning enabled real-time occupational safety monitoring in cold storage warehouse. *Saf Sci.* 2022;152:105766. doi: 10.1016/j.ssci.2022.105766
- [12] The Most Common Areas for Warehouse Dangers. American National Standard Institute. 2018. <https://blog.ansi.org/2018/06/most-common-areas-warehouse-dangers/#gref>
- [13] Hofstra N, Petkova B, Dullaert W, et al. Assessing and facilitating warehouse safety. *Saf Sci.* 2018;105:134–148. doi: 10.1016/j.ssci.2018.02.010
- [14] Gille C, Gunn E. A review of warehouse models. *Eur J Oper Res.* 1992;58(1):3–13. doi: 10.1016/0377-2217(92)90231-W
- [15] Zhang RQ, Wang M, Pan X. New model of the storage location assignment problem considering demand correlation pattern. *Comput Ind Eng.* 2019;129:210–219. doi: 10.1016/j.cie.2019.01.027
- [16] Petersen CG. The impact of routing and storage policies on warehouse efficiency. *Int J Oper Prod Manag.* 1999;19(10):1053–1064. doi: 10.1108/01443579910287073
- [17] Turner S. Joint Multi-item Storage Location Assignment Capacitated Lot Sizing Problem[master's thesis]. Windsor(ON): University of Windsor; 2009.
- [18] Gagliardi JP, Renaud J, Ruiz A. On storage assignment policies for unit-load automated storage and retrieval systems. *Int J Prod Res.* 2012;50(3):879–892. doi: 10.1080/00207543.2010.543939
- [19] Ramtin F, Pazour JA. Product allocation problem for an AS/RS with multiple in-the-aisle pick positions. *IIE Trans Institute Ind Eng.* 2015;47(12):1379–1396. doi: 10.1080/0740817X.2015.1027458

- [20] Zhang G, Nishi T, Turner SDO, et al. An integrated strategy for a production planning and warehouse layout problem: modeling and solution approaches. *Omega* (United Kingdom). 2017;68:85–94. doi: [10.1016/j.omega.2016.06.005](https://doi.org/10.1016/j.omega.2016.06.005)
- [21] Lesch V, Müller PBM, Krämer M, et al. Optimizing storage assignment, order picking, and their interaction in mezzanine warehouses. *Appl Intell*. 2023;53(15):18605–18629. doi: [10.1007/s10489-022-04443-x](https://doi.org/10.1007/s10489-022-04443-x)
- [22] Yang P, Miao L, Xue Z, et al. An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems. *J Intell Manuf*. 2015;26(6):1145–1159. doi: [10.1007/s10845-013-0846-7](https://doi.org/10.1007/s10845-013-0846-7)
- [23] Guerriero F, Musmanno R, Pisacane O, et al. A mathematical model for the multi-levels product allocation problem in a warehouse with compatibility constraints. *Appl Math Model*. 2013;37(6):4385–4398. doi: [10.1016/j.apm.2012.09.015](https://doi.org/10.1016/j.apm.2012.09.015)
- [24] Sanei O, Nasiri V, Marjani MR, et al. A heuristic algorithm for the warehouse space assignment problem considering operational constraints: with application in a case study. In: *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management*; Kuala Lumpur (Malaysia). 2011. 258–264.
- [25] Silva A, Coelho LC, Darvish M, et al. Integrating storage location and order picking problems in warehouse planning. *Transp Res Part E Logist Transp Rev*. 2020;140(April):102003. doi: [10.1016/j.tre.2020.102003](https://doi.org/10.1016/j.tre.2020.102003)
- [26] Kübler P, Glock CH, Bauernhansl T. A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses. *Comput Ind Eng*. 2020;147:106645. doi: [10.1016/j.cie.2020.106645](https://doi.org/10.1016/j.cie.2020.106645)
- [27] Bortolini M, Botti L, Cascini A, et al. Unit-load storage assignment strategy for warehouses in seismic areas. *Comput Ind Eng*. 2015;87:481–490. doi: [10.1016/j.cie.2015.05.023](https://doi.org/10.1016/j.cie.2015.05.023)
- [28] Larco JA, de Koster R, Roodbergen KJ, et al. Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions. *Int J Prod Res*. 2017;55(21):6407–6422. doi: [10.1080/00207543.2016.1165880](https://doi.org/10.1080/00207543.2016.1165880)
- [29] Heragu SS, Du L, Mantel RJ, et al. Mathematical model for warehouse design and product allocation. *Int J Prod Res*. 2005;43(2):327–338. doi: [10.1080/00207540412331285841](https://doi.org/10.1080/00207540412331285841)
- [30] Zaerpour N, De Koster RBM, Yu Y. Storage policies and optimal shape of a storage system. *Int J Prod Res*. 2013;51(23–24):6891–6899. doi: [10.1080/00207543.2013.774502](https://doi.org/10.1080/00207543.2013.774502)
- [31] Kovács A. Optimizing the storage assignment in a warehouse served by milkrun logistics. *Int J Prod Econ*. 2011;133(1):312–318. doi: [10.1016/j.ijpe.2009.10.028](https://doi.org/10.1016/j.ijpe.2009.10.028)
- [32] Borangiu T, Trentesaux D, Thomas A, et al. Adaptive storage location assignment for warehouses using intelligent products. *Stud Comput Intell*. 2016;271–279. doi: [10.1007/978-3-319-15159-5\\_25](https://doi.org/10.1007/978-3-319-15159-5_25)
- [33] Mirzaei M, Zaerpour N, de Koster R. The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance. *Transp Res Part E Logist Transp Rev*. 2021;146:146. doi: [10.1016/j.tre.2020.102207](https://doi.org/10.1016/j.tre.2020.102207)
- [34] Li J, Moghaddam M, Nof SY. Dynamic storage assignment with product affinity and ABC classification—a case study. *Int J Adv Manuf Technol*. 2016;84(9–12):2179–2194. doi: [10.1007/s00170-015-7806-7](https://doi.org/10.1007/s00170-015-7806-7)
- [35] Chiang DMH, Lin CP, Chen MC. The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterp Inf Syst*. 2011;5(2):219–234. doi: [10.1080/17517575.2010.537784](https://doi.org/10.1080/17517575.2010.537784)
- [36] Ming-Huang Chiang D, Lin CP, Chen MC. Data mining based storage assignment heuristics for travel distance reduction. *Expert Syst*. 2014;31(1):81–90. doi: [10.1111/exsy.12006](https://doi.org/10.1111/exsy.12006)
- [37] Walter R, Boysen N, Scholl A. The discrete forward-reserve problem - allocating space, selecting products, and area sizing in forward order picking. *Eur J Oper Res*. 2013;229(3):585–594. doi: [10.1016/j.ejor.2013.02.047](https://doi.org/10.1016/j.ejor.2013.02.047)
- [38] Chou YC, Chen YH, Chen HM. Recency-based storage assignment and warehouse configuration for recurrent demands. *Comput Ind Eng*. 2012;62(4):880–889. doi: [10.1016/j.cie.2011.12.009](https://doi.org/10.1016/j.cie.2011.12.009)
- [39] Choy KL, Ho GTS, Lee CKH. A RFID-based storage assignment system for enhancing the efficiency of order picking. *J Intell Manuf*. 2017;28(1):111–129. doi: [10.1007/s10845-014-0965-9](https://doi.org/10.1007/s10845-014-0965-9)
- [40] Yang L, Xu Y, Zheng Y, et al. Research on location assignment model of intelligent warehouse with RFID and improved particle swarm optimization algorithm. 2017. *Int Conf Comput Syst Electron Control ICCSEC 2017*. 2018;1262–1266. doi: [10.1109/ICCSEC.2017.8446952](https://doi.org/10.1109/ICCSEC.2017.8446952)
- [41] Courtin P, Grimault A, Lhommeau M, et al. Dynamic storage location assignment problem at Knapp company : toward reinforcement learning? Problematic and state of the art. Angers, Fr: *Artif Intell Platform*; 2020.
- [42] Tabatabaei S-K, Valilai OF, Abedian A, et al. A novel framework for storage assignment optimization inspired by finite element method. *PeerJ Comput Sci*. 2021;7:e378. doi: [10.7717/peerj-cs.378](https://doi.org/10.7717/peerj-cs.378)
- [43] Wachter JK, Yorio PL. A system of safety management practices and worker engagement for reducing and preventing accidents: an empirical and theoretical investigation. *Accid Anal Prev*. 2014;68:117–130. doi: [10.1016/j.aap.2013.07.029](https://doi.org/10.1016/j.aap.2013.07.029)
- [44] de Koster R, Le-Duc T, Roodbergen KJ. Design and control of warehouse order picking: a literature review. *Eur J Oper Res*. 2007;182(2):481–501. doi: [10.1016/j.ejor.2006.07.009](https://doi.org/10.1016/j.ejor.2006.07.009)
- [45] Durdevic D, Andrejic M, Pavlov N. Framework for improving warehouse safety. 5th *Logist Int Conf*. 2022;304–313. doi: [10.37528/FTTE/9788673954530.LO](https://doi.org/10.37528/FTTE/9788673954530.LO)
- [46] Kostrzewski M. Securing of safety by monitoring of technical parameters in warehouse racks, in high-bay warehouses and high storage warehouses – literature review of the problem. *LogForum*. 2017;13(2):125–134. doi: [10.17270/J.LOG.2017.2.1](https://doi.org/10.17270/J.LOG.2017.2.1)
- [47] Downie B, Gyöngyösi M, Kuehl C. Artificial intelligence in materials handling: how machine learning tools boost warehouse safety, productivity and cost-effectiveness. *J Supply Chain Manag Logist Procure*. 2021;4(1):6–16.
- [48] Gu J, Goetschalckx M, McGinnis LF. Research on warehouse operation: a comprehensive review. *Eur J Oper Res*. 2007;177(1):1–21. doi: [10.1016/j.ejor.2006.02.025](https://doi.org/10.1016/j.ejor.2006.02.025)
- [49] WorkSafe Victoria. A guide to handling large, bulky or awkward items. Australia; 2012.

- [50] WorkSafe Victoria. Pallet racking operation and maintenance. Australia; 2012.
- [51] Bartholdi J, Hankman S. Warehouse & distribution Science Release 0.98. Vol. 25. Atlanta (GA): Supply Chain Logist Inst; 2019.
- [52] Wales DJ, Doye JPK. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J Phys Chem A*. 1997;101(28):5111–5116. doi: 10.1021/jp970984n
- [53] Xiang Y, Gubian S, Suomela B, et al. Generalized simulated annealing for global optimization: the GenSA package. *R J*. 2013;5(1):13–28. doi: 10.32614/RJ-2013-002
- [54] Ben-Ameur W. Computing the initial temperature of simulated annealing. *Comput Optim Appl*. 2004;29(3):369–385. doi: 10.1023/B:COAP.0000044187.23143.bd
- [55] Zhao Y, Chen X, Li J. Tgmin: a global-minimum structure search program based on a constrained basin-hopping algorithm. *Nano Res*. 2017;10(10):3407–3420. doi: 10.1007/s12274-017-1553-z
- [56] Žunić E, Beširević A, Delalić S, et al. A generic approach for order picking optimization process in different warehouse layout. 41st International Convention On Information And Communication Technology, Electronics And Microelectronics (MIPRO). 2018;1000–1005. doi: 10.23919/MIPRO.2018.8400183
- [57] Bestuzheva K, Besançon M, Chen W-K, et al. The SCIP optimization suite 8.0. 2021. <https://optimization-online.org/?p=18429>
- [58] Perron L, Furnon V. OR-Tools v9.5. Google. 2022. Available from: <https://developers.google.com/optimization/>
- [59] Curtin K. Operations research. In *Encyclopedia of social measurement*. Elsevier; 2005. pp. 925–931. doi:10.1016/B0-12-369398-5/00534-X.
- [60] Alrawahi AS, Lee K, Lotfi A. A multiobjective QoS model for trading cloud of Things Resources. *IEEE Internet Things J*. 2019;6(6):9447–9463. doi: 10.1109/JIOT.2019.2943284
- [61] Zarinchang A, Faghieh N, Zarinchang J. An application of genetic algorithm toward solving the reliability problem of multiobjective series-parallel systems. *Eksplot i Niezawodn*. 2012;14(3):243–248.