ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 8, pp. 285-298
DOI:10.33168/JSMS.2024.0818

Optimising Warehouse Order Picking Through An Integrated Harmony Search and Simulated Annealing Algorithm: Application For A Food Packaging Manufacturer

Phornprom Rungrueang [1], Nattapol Paisarnvirosrak [2*]

[1] Faculty of Management Sciences, Kasetsart University, 199 Moo 6, Sukhumvit Road, Tung Sukla, Sri Racha, Chon Buri, 20230 Thailand

[2] Faculty of Logistics and Digital Supply Chain, Naresuan University, 99 Moo 9 Phitsanulok-Nakhonsawan Road, Mueang Phitsanulok District, Phitsanulok, 65000 Thailand

nattapolpa@nu.ac.th (Corresponding author)

Abstract. Nowadays, logistics is becoming highly competitive to increase profit and provide improved customer services. Efficient logistics management reduces business costs, providing competitive customer service. Therefore, the optimal storage location assignment could provide an effective solution such as maximises the performance of the order-picking process by reducing lead times and total storage travelling distances. This paper presented a novel integrated metaheuristics approach by combining harmony search (HS) and simulated annealing (SA) algorithms to optimise storage assignment in a food packaging warehouse and minimise order-picking time. The proposed HSSA algorithm reduced average route duration by 77.21% and cut total annual labour costs by 83.03% compared with the current layout, generating an effective storage location procedure with labour productivity improved by 335.85%. The revised configuration also reduced total carbon emissions from 9,160 to 7,327 $kgCO_2$.

Keywords: Storage Allocation, Warehouse Management, Metaheuristics, Warehouse Optimisation, Order-picking, Warehouse planning

1. Introduction

Recent rapid changes in supply chain management systems have encouraged businesses to increase customer satisfaction by minimising delivery time while also reducing inventory levels for cost minimisation. Well-organised warehouse management has become a crucial logistics challenge for overall supply chain efficiency. Warehouses are used for buffering and storage of many types of goods to guarantee the quantities demanded by customers in the shortest possible time (Kübler et al., 2020). The main processes of warehouse operations consist of reception, storage, order-picking and dispatch. Decision-making based on location for accommodating goods and the picking process (Storage Location Assignment Problem: SLAP) impacts the operational efficiency of warehouse space utilisation, cycle time for order preparation and order-picking operations (Reyes et al., 2019).

The order-picking process incurs the highest costs of warehouse operations (Manzini et al., 2015), estimated to be 55% of warehouse operating costs (Koster et al., 2007). Well-organised storage location assignment reduces item retrieval time, while low order lead times mitigate total travelling distances, enhance product flow and harmonise the different warehouse activities (Wisittipanich and Kasemset, 2015). By contrast, poorly organised storage location assignment results in a longer order-picking process, higher labour costs and reduced levels of customer satisfaction (De Koster et al., 2017).

Thus, here, a metaheuristics-based harmony search (HS) algorithm was integrated with a simulated annealing (SA) algorithm to solve the storage location assignment problem (SLAP). The main objective of this paper was to determine an optimal location assignment for each item of a food packaging manufacturer by considering total pickup minimisation time in an order-picking process.

1.1. Problem background

This paper focused on the planning phase for storage location assignment of a manually operated food packing manufacturing warehouse. In this case study, four pickers and one forklift driver were assigned to order-picking duty. The pickers picked the food packing based on customer orders of the type and quantity of each product. The food packaging warehouse consisted of 39 SKUs (Stock Keeping Units) such as bags, boxes, trays and wrappers. SKUs are used by retailers to identify and track inventory, combined with product characteristics such as price, product details and manufacturer details.

1.2. Research question

The goal of this paper was to discover the effectiveness of HSSA algorithms when solving the SLAP to minimise total pickup time during the order-picking process.

1.3. Contributions

This paper considered the SLAP for picker staff in the food packing warehouse and assumed that historical orders were predetermined and known. The three main contributions of this paper are listed below:

(1) This is the first paper that combines the HS with SA algorithms to solve the SLAP by minimising order-picking time. The HSSA provided a reasonable solution quality to complement and balance the intensification and diversification of the algorithm. HS randomly selected solution candidates, based on the efficiency of the algorithm, while SA helped to avoid traps in a local optimal search space, giving competitive results.

(2) The HSSA algorithm obtained reasonable quality solutions with a short computation time.

(3) The food packaging warehouse had poorly organised storage locations, making items difficult to find. Picker staff movement was reduced by reallocating items to more appropriate locations. Optimal routing was provided for staff (pickers) to minimise total order-picking time, number of workers, total labour costs and electricity consumption. Item reallocation was simpler than rearranging the layout which would have incurred high investment costs.

2. Literature review

This section describes the theoretical background of the methodology for solving the SLAP. Relevant overviews of particular areas of study are presented to gain a more comprehensive understanding of the existing research covering the HS and SA algorithms.

2.1.Storage location assignment problem (SLAP)

The SLAP is categorised as an NP-hard problem. Various types of items in a supply chain system must be stored in a warehouse that has limited storage capacity. Different warehouse designs cater to the huge number of items, arrival times and product demand uncertainty. Inefficient storage planning leads to higher costs of material handling, with poor space utilisation and higher cycle times of order preparation and order picking (Gu et al., 2010). Therefore, the SLAP must be taken into consideration.

In warehouse management, order-picking is one of the most time-consuming and labour-intensive processes for retrieving products from storage locations based on customer orders, representing 35% of the total expenses (Koster et al., 2007). Therefore, assigning a proper storage location to each type of product is a crucial task for warehouse management planning. Minimising order-picking time can greatly reduce warehouse operational expenses and product delivery times (Wang & Zhang, 2019).

Furthermore, items placed in storage locations must be rotated following the rotation patterns of each item and warehouse policies. Several warehouse policies are utilised to allocate items in a proper location such as random storage, dedicated storage and class-based storage (Hausman et al., 1976). Random storage involves random locations for item allocation, while in dedicated storage each type of product has a pre-assigned location (fixed location). Class-based storage is incorporated with location assignment based on physical item classification such as value, size and hazardousness. The closest free space and inventory rotation were introduced by Gómez et al. (2008). The closest free space policy relates to products that have prioritised storage at shorter distance locations such as near the reception or dispatching areas. The inventory rotation policy involves product-allocated sales indices. Products with higher sales turnover are stored near the reception or dispatching areas for easy access.

Several methodologies have been proposed based on operations research (OR) techniques to solve the SLAP, with metaheuristics methods. Xie et al. (2015) developed the restricted neighbourhood Tabu search (TS) to solve the SLAP by considering grouping constraints. Their solution was optimal for small-scale instances to minimise operational costs. Yang et al. (2015) established the variable neighbourhood search (VNS) algorithm that considered shared storage and retrieval scheduling for multi-shuttle automated storage and retrieval systems to solve the SLAP, with results displaying a gradual reduction in algorithm running time, while Zhang et al. (2019) proposed a genetic algorithm (GA) to solve the SLAP in an order-picking system using a case study of mobile rack warehouses. Their proposed algorithm provided a better solution than random storage assignment strategy (RAS) and good-clustering storage location assignment strategy (GCAS). Mendes et al. (2022) implemented an iterated local search (ILS) with a SLAP case study for a pharmaceutical product distributor with isolation constraints. Their proposed algorithm efficiently handled the SLAP for large data with reasonable running time.

Previous studies investigated the SLAP with the main objective to optimise travelling distance considering only horizontal storage. An HSSA algorithm has never been developed to solve the SLAP. To fill the current research lacuna, this HSSA algorithm was developed to resolve the SLAP by considering both vertical and horizontal storage allocations. The major objective of this research focused on minimising total order-picking time by providing order-picking routes and item reallocations to the pickers. In this research, total order-picking time refers to the time required to execute the order-picking process. The three components of total order-picking time include walking time (travelling time), picking time and dead time. Walking time refers to the time taken by the picker to reach the item location and return to the packing zone. Picking time is the time required for the actual order-picking process. Dead unproductive time involves searching, reading instructions and checking for picking article locations.

2.2.Harmony search algorithm

Currently, many methodologies are utilised to optimise real-life problems. The harmony search (HS) metaheuristics-based algorithm was developed by Geem et al. (2001) to solve combinatorial optimisation problems. HS algorithm development was influenced by the natural improvement process of a musician for pitch adjustment to attain better harmony. This process follows similar methods to the searching procedures of local and global search. The characteristics of HS can be divided into four basic stages (Yi et al., 2019). In the first stage, the initial harmony is randomly generated before being sent to the harmony memory. Next, in the improvisation stage, a new candidate harmony is developed. During the improvisation stage, three main rules of generating new harmony vectors draw on random selection, harmony memory consideration and pitch adjustment. If a new candidate harmony in the harmony memory is presented with an improvement of the harmony vector, the worst harmony vector is updated and swapped with the new candidate in the third stage. Then, this procedure is repeated until a stopping condition is met in the fourth stage. The procedure will be repeated in the second stage if the stop condition is not met (Yassen et al., 2013).

To the best of our knowledge, no reviews have utilised the HS algorithm to solve the SLAP. Compared with other metaheuristics, the HS algorithm has fewer parameters and generates more robustness to handle several problems. The HS algorithm is efficient with simple implementation and is popularly proposed for managing various logistics and supply chain problems The HS algorithm has been used to solve scheduling problems such as flow shop scheduling (Doush et al., 2022), nurse rostering problems (Hadwan, 2022) and vehicle routing problems (Chen et al., 2017; Maleki et al., 2017; Yassen et al., 2013). It has also been applied to deal with industrial problems such as load dispatch problems (Karthigeyan et al., 2015) and optimal reactive power flow problems (Sivasubramani et al., 2011).

Advantages of the HS algorithm are that decision variables do not need to be initialised and also few control parameters are required to perform better solutions. HS is also simple to apply for solving optimisation problems. Furthermore, the derivative information is not obliged. Promising solutions were provided by the HS algorithm (Qin et al., 2022). The HS algorithm has many advantages but the imbalance between global and local search has become one crucial drawback, resulting in low solution quality. To enhance performance, many researchers have integrated the HS algorithm with other parameters (Yassen et al., 2015). However, recently, the HS algorithm has presented unfavourable metaheuristics for researchers.

2.3.Simulated annealing algorithm

Kirkpatrick et al. (1983) introduced the simulated annealing (SA) algorithm to deal with combinatorial optimisation problems. SA algorithm development was influenced by the process of metal annealing as the material heating and cooling processes to improve ductility. SA is a metaheuristics algorithm that is simple to implement for solving several kinds of problems (Streijffert & Tegelmark, 2020). The SA algorithm obtains non-improving solutions to lead to better results, with an improved ability to search in larger portions of the candidate solution, while avoiding the local optimal (Santosa & Kresna, 2015).

Many studies have been conducted to optimise SA but few have utilised SA to solve the SLAP. Muppani and Adil (2008) studied an SA algorithm to solve class formation and storage assignment using an integer programming model. They evaluated all possible products, with the main objectives of optimising storage space and order-picking costs using a mathematical model and the SA algorithm to solve the SLAP by basic dedicated storage strategy. Their paper minimised the cost of storage allocation, order-picking costs and storage costs. The SA algorithm can reach a diverse solution with reasonable cost reduction (Atmaca & Ozturk, 2013). Mirzaei et al. (2021) proposed an integrated cluster allocation (ICA) algorithm by considering product affinity and turnover. A greedy construction heuristic was developed to solve the SLAP as a large-sized problem, resulting in a significant retrieval time reduction of up to 40%. Yuan et al. (2021) studied the storage assignment optimisation of robotic mobile fulfilment system (RMFS) by considering the assignment stage and pods assignment stage. They developed a two-stage hybrid algorithm that integrated a greedy algorithm and simulated annealing which achieved reasonable reductions in order-picking time.

One major advantage of the SA algorithm is that it can avoid becoming trapped in a local optimal. When the worst solution candidate was accepted, the algorithm randomly selected a better solution candidate. Another significant advantage of the SA algorithm is that it provides a flexible and robust framework to handle constraints. The SA algorithm does not rely on any restrictive model properties. Therefore, it is simple and easy to implement and has the versatility to find the optimal solution. However, one crucial disadvantage of the SA algorithm is the search process which requires a long computation time (Busetti, 2003).

3. Methodology

This section describes the problem description, the HS framework that integrates SA, the mathematical model of HSSA, the pseudo code of HSSA for solving the SLAP.

3.1.Description of the problem

This Storage Location Assignment Problem (SLAP) case study had 66 storage location shelves with 5 levels of pallet racks. The storage warehouse area was 120,000 square metres and the weight capacity of each trip could not exceed 2,000 kilograms. There were 39 SKUs and 4 full-time employees in the order-picking area.

For the proposed methodology we set HMS = 30, HMCR = 0.8 and PAR = 0.6 following Chen et al. (2017) who integrated HS with VNS using a similar sample size to our case study. To integrate HS with SA, we set the parameter of the SA initial temperature *temp* = 0.95 and cooling function factor is α = 0.90 based on Attiya & Hamam (2006) because many successful cases in the literature use these values.

The problem involves storage location assignment a set of *M* items into a set of *N* locations. The routing network for pick up the item in warehouse is represented by a graph *G* = (V, A), where *V* = {0, 1, …, *N*} is the set of nodes and *A* = {(*i*, *j*) | *i*, *j* ∈ V, *i* ≠ *j*} is the set of arcs which connect between node *i* and node *j*. Node 0 represents the outbound station used for starting and ending the delivery trip and the other nodes (except 0) indicate item locations. For each item *m* (1 ≤ *m* ≤ *M*), the total stock of each item Q_m is known.

For each delivery trip of order number *k* (1 ≤ *k* ≤ *K*), the capacity *L* is given. The pickup demand of each item m from storage P_m^k and demand of each order *k* D_m^k are shown in the received order. In addition, q_i which is the number of item in each location *i* also shown.

Delivery time C_{mij} of item *m* between location *i* to *j* is associated with each arc (i, j) ∈ A and both are asymmetric delivery times. The decision variables are denoted by X_{mij}^k which is a binary number and $i, j \in N$ where X_{mij}^k is equal to 1 when the delivery trip on order number k, receives the item *m* and travels from location *i* to *j*; otherwise X_{mij}^k is equal to 0. Moreover, Y_{mi} which is a binary number and *i* ∈ *N* where Y_{mi} is equal to 1 when item m was assigned to location *i;* otherwise Y_{mi} equal to 0. The warehouse layout of the case study is displayed in Figure 1.

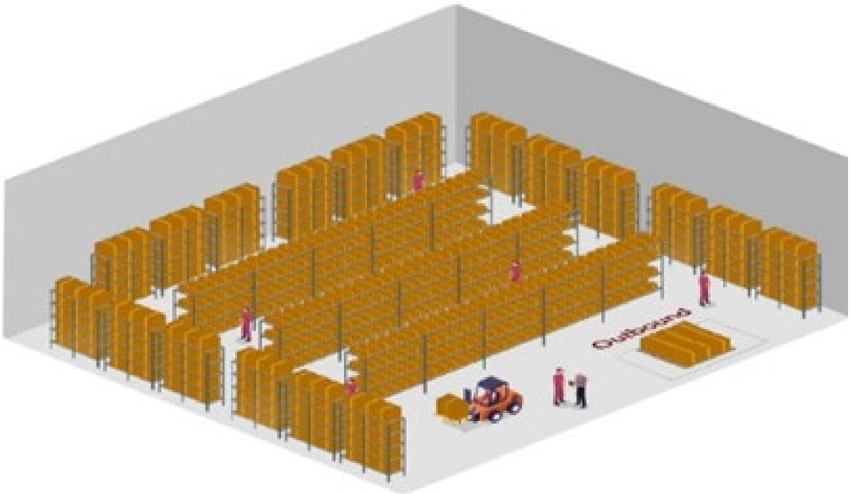

Fig. 1: Warehouse layout

3.2.Model formulation

The main aim of this paper was to minimise the total order-picking process time. Formulation of the SLAP was presented as follows.

Objective function:

$$\text{Minimise } Z = \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} X_{mij}^{k} C_{mij} \tag{1}$$

Subject to:

$$\sum_{m=1}^{M} Y_{mi} = 1 \tag{2}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=0}^{N} X_{m0j}^{k} \geq K \tag{3}$$

$$\sum_{j=0}^{N} X_{m0j}^{k} - \sum_{i=0}^{N} X_{mi0}^{k} = 0;\ \text{for } k = \{1, 2, \dots, K\} \tag{4}$$

$$\sum_{m=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} X_{mij}^{k} P_{m}^{k} \geq D_{m}^{k};\ \text{for } k = \{1, 2, \dots, K\} \tag{5}$$

$$\sum_{i=1}^{N} Y_{mi} q_i = Q_m \quad ;\ \text{for } m = \{1, 2, \dots, M\} \tag{6}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} X_{mij}^{k} P_{m}^{k} \leq \sum_{m=1}^{M}\sum_{i=1}^{N} Y_{mi}\, q_i \tag{7}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} X_{mij}^{k} P_{m}^{k} \leq L \tag{8}$$

$$X_{mij}^{k} \in \{0,1\};\quad i, j \in N;\quad i \neq j;\quad m \in M;\quad k \in K \tag{9}$$

$$Y_{mi} \in \{0,1\};\quad i \in N;\quad m \in M \tag{10}$$

Equation (1) shows the objective function of the problem to minimise the total order-picking time. Equation (2) indicates that the storage can hold only one item at one location. Equation (3) indicates that every pickup order should be satisfied and received from the warehouse. Equation (4) states that every pickup trip should start from and return to the outbound station. Equation (5) ensures that the demand of each order has been met. Equation (6) confirms that the item in all storage location should be match with total stock of the item. Equation (7) checks that the item stocks in the warehouse will be sufficient to cover the all order. Equation (8) confirms that the total pickup item cannot exceed the trip capacity. Finally, Equation (9) and (10) gives the decision variables defined as binary numbers.

3.3.Pseudo code of HSSA algorithm

The pseudo code of the HSSA algorithm is displayed in Figure 2. This pseudo code provides the working process of the Integrated Harmony Search Algorithm with Simulated Annealing (HSSA), The algorithm can be divided into five main steps which are initialisation (lines 01 to 08), checking the termination criterion (line 09), improving a new solution from the Harmony Memory (HM) (lines 10 to 20), updating the HM and setting an exception criterion using the Simulated Annealing (SA) algorithm (lines 21 to 29) and providing the optimal solution (line 32).

```
Pseudo code of the Integrated Harmony Search Algorithm with Simulated Annealing (HSSA)
-------------------------------------------------------------------------------------------------
01: Pseudo code of the HSSA (t, r_h , r_s, a, Temp, i, PAR, HMCR, HM)
                                        ▻ t : number of iterations
                                        ▻ r_h : random number between 0 and 1
                                        ▻ r_s : random number between 0 and 1
                                        ▻ M : number of items
                                        ▻ α : Cooling factor
                                        ▻ PAR : Pitch adjusting rate
                                        ▻ HMCR : Harmony Memory Considering rate
                                        ▻ HM : Harmony Memory
                                        ▻ Temp : Temperature
02: Set all parameters t = 1, i = 1
03: Set Harmony Memory Considering rate (HMCR)
04: Set Pitch adjusting rate (PAR)
05: Set the objective function f(x)
06: Set a(Temp) as a cooling function;
07: Set initial temperature Temp > 0
08: Generate the initial solution from Harmony Memory with random harmonies solution (s');
09:   while (t < t^max) do
10:       for (i  = 1 to M) do
11:           if (r_h < HMCR) then
12:               Select value from HM;
13:               if (r_h < PAR) then
14:                   Adjust the value of pick-up order;
15:               end if
16:           else
17:               Choose a random value;
18:           end if
19:         i++;
20:       end for
21:       Δf = worst solution (s^HM) - solution (s^ans);
22:       if solution (s^ans) < worst solution (s^HM) then
23:            Replacement solution (s^ans) in HM;
24:       else if (exp (-Δf)/Temp) > r_s) then
25:             Replacement solution (s^ans) in HM;
26:       else
27:           return HM;
28:       end if
29:       Set Temp = a(Temp);
30:       t++;
31:   end while
32:   Return solution (s^best) in HM as optimal solution;
33: end procedure
```

Fig. 2: Pseudo code of the HSSA algorithm

The initial parameters, initial solution and stopping criterion must be set before starting the algorithm. The new solution is generated by selection from the HM. This solution is improved by following the Harmony Search (HS) procedure. The SA algorithm is then modified into the HS algorithm to avoid trapping in the local optimal that can occur when using neighbourhood search. Finally, if the new solution presents a result better than the worst solution, the HM will be updated. After the stopping criterion is met, the best solution in the HM is the optimal solution of the HSSA algorithm.

4. Results and Discussions

This section we present the result of the proposed methodology describe in section 3 for SLAP solution method, analyses the performance of HSSA algorithm, the result discussions and the validation of HSSA algorithm.

The section implements the proposed HSSA algorithm to deal with the SLAP as a case study of a food packaging manufacturer. The proposed HSSA algorithm was coded in JavaScript on a CPU Intel® Core™ i5-5200U, up to 2.7 GHz with 4 GB of RAM. The stopping criterion of the proposed algorithm was based on 25 minutes running time recommended by Yassen et al. (2015), The computational experimentation results using the HSSA algorithm were compared with the currently used warehouse layout, as presented in Figure 3.

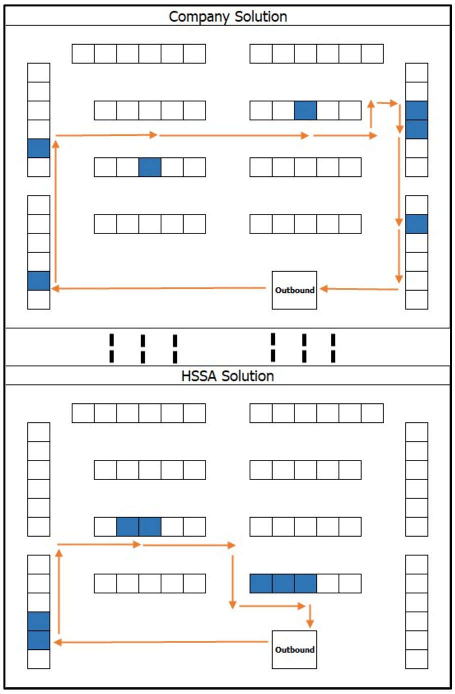

Fig. 3: An example of order picking routing by the proposed HSSA algorithm compared with the current company solution

The company warehouse used randomisation for item collection, with four workers operating in the order-picking area. The experimentation was conducted using ten company orders. Results in Table 1 compare the order-picking time of the proposed HSSA algorithm, the HS algorithm and the company solution. The proposed combined HSSA algorithm reduced average order-picking time to 25.96 min, with a 77.21% improvement compared to the company solution. Order-picking time also reduced by 11.29% compared to the HS algorithm.

Table. 1: Comparison of order-picking time between HSSA, company solution, SA solution and HS solution

RECEIPT NO.	***ORDER-PICKING TIME (MIN)***			
	Company solution	**SA solution**	**HS solution**	**HSSA solution**
1	96.45	38.27	26.54	22.48
2	98.54	36.28	27.24	26.35
3	110.11	32.68	28.46	25.31
4	99.54	38.51	27.51	26.5
5	104.48	34.54	28.35	26.67
6	97.19	36.18	26.57	25.29
7	102.46	36.24	37.24	26.55
8	141.59	36.54	26.02	24.59
9	147.16	39.36	31.23	30.78
10	141.31	42.43	33.46	25.05
AVERAGE	113.88	36.80	29.26	25.96

The proposed HSSA algorithm generated the most efficient solution for order-picking time minimisation as the main aim of this paper. The HSSA algorithm results are compared with the company solution in Tables 2, 3 and 4, showing reduced annual total working time, operating expenses and carbon emissions respectively.

Table. 2: Comparison of total working time between the two solutions of the HSSA algorithm (V1 and V2) and the company solution

	Company solution	*HSSA solution (V1)*	*HSSA solution (V2)*
No. of workers	4	1	2
Working time (per day)	8 hours	8 hours	8 hours
Working time (per year)	8,448 hours	2,112 hours	4,224 hours
OT (per year)	1,232 hours	95 hours	-
Total working time (per year)	9,680 hours	2,207 hours	4,224 hours

Total working time of employees in the order-picking department was compared between the proposed HSSA algorithm and the company solution (Table 2), All workers can work overtime (OT) for only 7 months per year. The OT payment rate was 150 Thai Baht per hour (Ministry of Labour, 2022), Total working time per year of the company solution in Table 2 includes both normal working time and OT. To illustrate the algorithm performance, two solutions were presented as HSSA solution (V1) and HSSA solution (V2). For HSSA solution (V1), only one picker was assigned in the order-picking area. This reduced the total working time per year from 9,680 to 2,207 hours. For solution V2, the number of picker staff was reduced by 2. Compared to HSSA (V1), the solution of HSSA (V2) did not require any overtime to complete the job. The V1 solution minimised the total annual working time. The total annual working time of HSSA (V2) was greater than HSSA solution (V1) but V2 provided an effective workload balance among staff pickers.

When the items were stored appropriately and the effective order-picking routes were followed, this minimised order-picking time and travelling distance, with reduced waste movement of pickers leading to reduced working hours.

Table. 3: Comparison of labour costs between the two solutions of the HSSA algorithm (V1 and V2) and the company solution

	Company solution	*HSSA solution (V1)*	*HSSA solution (V2)*
No. of workers	4	1	2
Salary (Thai Baht per year)	372,768	93,192	186,384
OT expense (Thai Baht per year)	184,800	14,250	-
Total labour cost (Thai Baht per year)	557,568	94,642	186,384

Table 3 compares the annual company labour costs between the company currently used warehouse layout and the two HSSA algorithm solutions. The HSSA algorithm reduced the number of pickers, leading to a large reduction of total labour cost per year. HSSA (V1) demonstrated a large reduction in annual OT expenses from 184,800 to 14,250 THB because all picking tasks were completed within normal working hours. The HSSA (V1) solution minimised total labour costs from 557,568 to 94,642 THB per year. HSSA (V2) reduced fatigue and minimised employee stress by assigning 2 pickers in the order-picking area. This distributed the workload evenly and indicated that a higher workload could be managed without the need to increase staffing.

Table. 4: Comparison of carbon emissions between the two solutions of the HSSA algorithm (V1 and V2) and the company solution

	Company solution	***HSSA solution (V1)***	***HSSA solution (V2)***
Working time (per day)	10 hours	9 hours	8 hours
Total electricity usage (kWh)	11,550 units	10,395 units	9,240 units
Total electricity cost (Thai Baht)	46,200	41,580	36,960
Total carbon emission (kgCO₂)	9,160	8,243	7,327

Table 4 compares the total carbon emissions between the company solution and the two solutions of the HSSA algorithm. The unified calculation formula of carbon emission followed Ries et al. (2017). The electricity cost was calculated using the rates provided by the Provincial Electricity Authority (2023). The HSSA algorithm (V1) reduced daily working time by one hour, with 10% lower electricity consumption. The total cost of electricity was reduced from 46,200 to 41,580 Thai Baht per year, while total carbon emissions also declined by 10.01% due to the direct relationship between electricity usage and carbon emissions released through order-picking activities. When comparing the company solution and the HSSA (V2), daily working time was reduced by 2 hours, resulting in lower electricity consumption. The HSSA algorithm (V2) optimised total carbon emissions as 11.11% lower than V1 and 20.01% lower than the current company solution.

Labour productivities of the company solution, the SA algorithm, the HS algorithm and the HSSA algorithm are presented in Table 5. In this research, the case study warehouse employed four workers generating 9,680 annual working hours. Labour productivity is defined as output per hour worked. This was calculated by the model formulation of Attar et al. (2012) as follows:

$$Labour\ productivity\ = \frac{Working\ hours\ per\ year}{Number\ of\ orders\ per\ year}$$

Table. 5: Comparison of labour productivity between the company solution, the SA algorithm, the HS algorithm and the HSSA algorithm

RECEIPT NO.	***LABOUR PRODUCTIVITY (NO. OF ORDER PER HOUR)***			
	Company solution	**SA algorithm**	**HS algorithm**	**HSSA algorithm**
1	0.62	1.57	2.26	2.67
2	0.61	1.65	2.20	2.28
3	0.54	1.84	2.11	2.37
4	0.60	1.56	2.18	2.26
5	0.57	1.74	2.12	2.25
6	0.62	1.66	2.26	2.37
7	0.59	1.66	1.61	2.26
8	0.42	1.64	2.31	2.44
9	0.41	1.52	1.92	1.95
10	0.42	1.41	1.79	2.40
AVERAGE	0.53	1.62	2.05	2.31

Results in Table 5 showed that the HSSA algorithm improved average labour productivity by 335.85% from 0.53 to 2.31. The HS algorithm improved average labour productivity by 1.52 compared to the company solution and by 0.43 compared to the SA algorithm.

4.1.Comparison of the sample T-test

Results generated by the proposed HSSA algorithm were validated by the paired sample T-Test (Table

6). The sample size of each instance consisted of 10 orders ($N = 10$) with degree of freedom equal to 9 ($Df = N - 1$). The average and standard deviations of the result were calculated for each instance and compared to evaluate significant changes between the HSSA algorithm, the HS algorithm and the SA algorithm. A value of $p \leq 0.05$, was set as implying significant difference.

Table. 6: Paired sample T-test comparisons between the HS algorithm, the SA algorithm and the HSSA algorithm

Instance	*HSSA VS SA*	*P ≤ 0.05*	*HSSA VS HS*	*P ≤ 0.05*
HSP001	0.007	✓	0.015	✓
HSP002	0.000	✓	0.000	✓
HSP003	0.000	✓	0.000	✓
HSP004	0.017	✓	0.152	×
HSP005	0.000	✓	0.000	✓
HSP006	0.000	✓	0.000	✓
HSP007	0.023	✓	0.049	✓
HSP008	0.000	✓	0.000	✓
HSP009	0.000	✓	0.000	✓
HSP010	0.000	✓	0.018	✓

The p-values verified that the HSSA algorithm was statistically better than the HS algorithm in 9 out of 10 instances and better than the SA algorithm in 10 out of 10 instances (Table 6). Therefore, the proposed HSSA algorithm provided acceptable solution quality.

5. Conclusions

This paper presented the HSSA algorithm to solve the SLAP for a food packaging manufacturer. The main aim was to optimise location allocation for each item to minimise total order-picking pickup time. The HS metaheuristics algorithm is efficient, simple to use and contains few mathematical parameters. However, the HS algorithm has limited robustness, with difficulty finding an optimal solution in a search space. The HS algorithm was integrated with the SA algorithm to enhance its ability. Our proposed HSSA algorithm has never been previously used to solve the SLAP.

The experienced approach was conducted with case study of food packaging manufacturer. After implementation with the HSSA algorithm, the results were compared with the general HS algorithm, SA algorithm and the company solution. The HSSA algorithm solution was divided into two versions (V1 and V2) with different numbers of order-picking staff. There was only one order picker for the HSSA solution (V1) and two order-picking staff for the V2 version. Results of both versions of the HSSA solutions are shown in Table 3 with the decline in annual OT expenses and total annual labour costs. The first version (V1) of the HSSA algorithm reduced expenses more than the HSSA solution (V2) and the company solution by 49.22% and 83.03% respectively. The HSSA solution V2 displayed a significant reduction in annual OT expenses because the pickers completed their tasks within the normal working hours, with no requirement to pay for overtime expenses and also gave reduced electricity consumption. The HSSA algorithm solution (V2) reduced carbon emissions by 20.01%. Two staff were assigned to the order-picking area to maintain the fairness of the workload, with more time to deal with order uncertainties and avoid exhaustion among staff pickers. Maintaining a healthy work-life balance is essential for overall well-being and productivity. The HSSA algorithm enhanced the labour productivity of order picking per hour more than the HS algorithm by 289.18%, the SA algorithm by 206.96% and the company solution by 335.85%.

To sum up, the HSSA algorithm optimised warehouse storage with reduced order-picking time and expenses, showcasing the benefits of tailored metaheuristics that balanced intensification and diversification for quality solutions. Practical recommendations are provided around change management and data inputs to enable adoption. As food packaging producers contend with margin pressures, esteeming logistics planning and dynamic storage layouts can drive substantial time and cost economies. Future work should assess model robustness across wider product portfolios and storage constraint scenarios.

6. Future work

In this paper, inbound storage was not taken into account due to data limitations. This will be considered in future studies. The HSSA algorithm could also be applied to other types of warehouses such as crossdocking and consolidated warehouses. For process improvement, the HSSA algorithm could be integrated with augmented reality (AR) as an order picking routing for a manual picker, while demand forecasting could also be taken into account to improve warehouse space utilisation. Our proposed HSSA algorithm could be extended by implementation with SLAP case studies under dynamic conditions. The HSSA algorithm could be developed for dealing with stochastic SLAP case studies. This HSSA algorithm can also applied to resolve other combinatorial optimisation problems.

References

Atmaca, E. & Ozturk, A. (2013). Defining order picking policy: A storage assignment model and a simulated annealing solution in AS/RS systems. *Applied Mathematical Modelling*, 37(7), 5069–5079.

Attiya, G. & Hamam, Y. (2006). Task allocation for maximizing reliability of distributed systems: A simulated annealing approach. *Journal of parallel and Distributed Computing*, 66(10), 1259-1266.

Busetti, F. (2003). Simulated annealing overview (World Wide Web). www.geocities.com/francorbusetti/saweb.pdf, 1–10.

Chen, S., Chen, R. & Gao, J. (2017). A Modified Harmony Search Algorithm for Solving the Dynamic Vehicle Routing Problem with Time Windows. *Scientific Programming*, 2017.

De Koster, R. B. M., Johnson, A. L. & Roy, D. (2017). Warehouse design and management. *International Journal of Production Research*, 55(21), 6327–6330.

Doush, I. A., Al-Betar, M. A., Awadallah, M. A., Alyasseri, Z. A. A., Makhadmeh, S. N. & El-Abd, M. (2022). Island neighboring heuristics harmony search algorithm for flow shop scheduling with blocking. *Swarm and Evolutionary Computation*, 74, 101127.

Geem, Z. W., Kim, J. H. & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76(2), 60–68.

Gómez, A., Iglesias, M., De La Fuente, D. & Menéndez, M. (2008). Estado del arte en políticas de ubicación de productos en almacén. *Insights on Current Organization Engineering: 2nd International Conference on Industrial Engineering and Industrial Management, XII Congreso de Ingeniería de Organización, September 3-5 2008, Burgos, Spain, 2008*, 255–256.

Gu, J., Goetschalckx, M. & McGinnis, L. F. (2010). Solving the forward-reserve allocation problem in warehouse order picking systems. *Journal of the Operational Research Society*, 61(6), 1013–1021.

Hadwan, M. (2022). Annealed Harmony Search Algorithm for Nurse Rostering Problems. *Computers, Materials and Continua*, 71(3).

Hausman, W. H., Schwarz, L. B. & Graves, S. C. (1976). Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science*, 22(6), 629–638.

Karthigeyan, P., Raja, M. S., Hariharan, R., Prakash, S., Delibabu, S. & Gnanaselvam, R. (2015). Comparison of Harmony Search Algorithm, Improved Harmony Search Algorithm with Biogeography Based Optimization Algorithm for Solving Constrained Economic Load Dispatch Problems. *Procedia Technology*, 21, 611–618.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(459), 671–680.

Koster, R., Le-Duc, T. & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.

Kübler, P., Glock, C. H. & Bauernhansl, T. (2020). A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses. *Computers and Industrial Engineering*, 147(July), 106645.

Maleki, F., Yousefikhoshbakht, M. & Rahati, A. (2017). A Hybrid Self-adaptive Global Best Harmony Search Algorithm for the Vehicle Routing Problem with Time Windows. *BRAIN – Broad Research in Artificial Intelligence and Neuroscience*, 8(4), 65–84.

Manzini, R., Accorsi, R., Gamberi, M. & Penazzi, S. (2015). Modeling class-based storage assignment over life cycle picking patterns. *International Journal of Production Economics*, 170, 790–800. DOI: 10.1016/j.ijpe.2015.06.026

Mendes, N. F. M., Bolsi, B. & Iori, M. (2022). An Iterated Local Search for a Pharmaceutical Storage Location Assignment Problem with Product-cell Incompatibility and Isolation Constraints. *International Conference on Enterprise Information Systems, ICEIS - Proceedings*, 1(Iceis), 17–26.

Ministry of Labour. (2022). *Announcement of the Wage Committee on Minimum Wage Rate (No. 11),*

Mirzaei, M., Zaerpour, N. & de Koster, R. (2021). The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance. *Transportation Research Part E: Logistics and Transportation Review*, 146(July 2020).

Muppani, V. R. & Adil, G. K. (2008). Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach. *Omega*, 36(4), 609–618.

Provincial Electricity Authority. (2023). *Ft surcharge during January – April 2023 Ft surcharge during January - April 2023* (Issue April).

Reyes, J. J. R., Solano-Charris, E. L. & Montoya-Torres, J. R. (2019). The storage location assignment problem: A literature review. *International Journal of Industrial Engineering Computations*, 10(2), 199–224.

Richards, G. (2014). *Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse* (2nd ed.), Kogan Page.

Qin, F., Zain, A. M. & Zhou, K. Q. (2022). Harmony search algorithm and related variants: A systematic review. *Swarm and Evolutionary Computation*, 74(September 2021), 101-126.

Ries, J. M., Grosse, E. H. & Fichtinger, J. (2017). Environmental impact of warehousing: a scenario analysis for the United States. *International Journal of Production Research*, 55(21), 6485–6499.

Santosa, B. & Kresna, I. G. N. A. (2015). Simulated Annealing to Solve Single Stage Capacitated Warehouse Location Problem. *Procedia Manufacturing*, 4(Iess), 62–70.

Sivasubramani, S. & Swarup, K. S. (2011). Multi-objective harmony search algorithm for optimal power flow problem. *International Journal of Electrical Power and Energy Systems*, 33(3), 745–752.

Streijffert, N. & Tegelmark, F. (2020). A Comparative Study of Simulated Annealing and Self-Organising Map Batching for Solving the Order Batching Problem in Warehouses.

Wang, M. & Zhang, R. Q. (2019). A dynamic programming approach for storage location assignment planning problem. *Procedia CIRP*, 83(March), 513–516.

Wisittipanich, W. & Kasemset, C. (2015). Metaheuristics for warehouse storage location assignment problems. *Chiang Mai University Journal of Natural Sciences*, 14(4), 361–377.

Xie, J., Mei, Y., Ernst, A. T., Li, X. & Song, A. (2015). A Restricted Neighbourhood Tabu Search for Storage Location Assignment Problem. *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, *May*, 2805–2812.

Yang, P., Miao, L., Xue, Z. & Ye, B. (2015). Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems. *Transportation Research Part E: Logistics and Transportation Review*, 79, 164–177.

Yassen, E. T., Ayob, M., Ahmad Nazri, M. Z. & Ahmad, Z. (2013). Harmony search algorithm for vehicle routing problem with time windows. *Journal of Applied Sciences*, 13(4), 633–638.

Yassen, E. T., Ayob, M., Nazri, M. Z. A., and Sabar, N. R. (2015). Meta-harmony search algorithm for the vehicle routing problem with time windows. *Information Sciences*, 325, 140–158.

Yi, J., Lu, C. & Li, G. (2019). A literature review on latest developments of Harmony Search and its applications to intelligent manufacturing. *Mathematical Biosciences and Engineering*, 16(4), 2086–2117.

Yuan, R., Li, J., Wang, W., Dou, J., & Pan, L. (2021). Storage Assignment Optimization in Robotic Mobile Fulfillment Systems. *Complexity*, 2021.

Zhang, D., Si, Y., Tian, Z., Yin, L., Qiu, J. & Du, X. (2019). A genetic-algorithm based method for storage location assignments in mobile rack warehouses. *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, 1–6.