A variable neighbourhood search for minimization of operation times through warehouse layout optimization

JON DÍAZ*, *DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain.*

HAIZEA RODRIGUEZ**, *DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain.*

JENNY FAJARDO-CALDERÍN†, *DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain.*

IGNACIO ANGULO††, *DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain.*

ENRIQUE ONIEVA§, *DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain.*

Abstract

For companies involved in the supply chain, proper warehousing management is crucial. Warehouse layout arrangement and operation play a critical role in a company's ability to maintain and improve its competitiveness. Reducing costs and increasing efficiency are two of the most crucial warehousing goals. Deciding on the best warehouse layout is a remarkable optimization problem. This paper uses an optimization method to set bin allocations within an automated warehouse with particular characteristics. The warehouse's initial layout and the automated platforms limit the search and define the time required to move goods within the warehouse. With the help of historical data and the definition of the time needed to move goods, a mathematical model of warehouse operation was created. An optimization procedure based on the well-known Variable Neighbourhood Search algorithm is defined and applied to the problem. Experimental results demonstrate increments in the efficiency of warehousing operations.

Keywords: Warehouse layout design, optimization of warehouse, decision support models, logistics, optimization, variable neighbourhood search

*E-mail: jondi@opendeusto.es
**E-mail: haizearodriguez@opendeusto.es
†E-mail: fajardo.jenny@deusto.es
††E-mail: ignacio.angulo@deusto.es
§E-mail: enrique.onieva@deusto.es

Vol. 32, No. 4,
Advance Access published 26 March 2024 https://doi.org/10.1093/jigpal/jzae018

1 Introduction

One of the main goals of global market trends is to make a real effort to get goods to customers quickly and, as a result, lowering the price of item storage. Although connecting the supplier and the customer directly has been considered, it has yet to become a reality, according to [3]. Companies still require a warehouse where goods can be stored and organized efficiently before distribution, and intelligent warehouses are the answer.

Over time, warehouses have evolved. Initially, all product handling was done manually, but in an effort to save time and money, technology was used to automate the main processes. The warehouse machinery has proven to be beneficial in terms of lowering the cost and increasing the efficiency of the movements. Nevertheless, some challenges are unachievable for even the most advanced machines. The proper arrangement of goods is one of the main obstacles encountered at warehouses when trying to optimize the services [1].

Warehouses used to have a fixed layout that was not meant to change, but how products are organized can be adjusted to meet the needs of the business. The main goal is to save time and bring the most frequently requested items closer to the extraction points, and nearly locating pairs of products extracted together often can help. In addition, because these products are constantly moving, reducing the effort required to access them lowers the overall cost. This has a similar effect in both manually and automatically managed warehouses.

Optimization techniques are critical to finding the best arrangement for the goods. The algorithms' solution can help lower costs and improve overall performance, but it can't guarantee that a perfect solution will be found. As a result, several algorithms are put to the test in order to find the arrangement that produces the best warehouse layout.

This article discusses a practical application of optimization methods, such as combinatorial algorithms to the storage of goods in a warehouse with automatic moving platforms. The goal of the optimization procedure is to reduce the time platforms spend moving goods internally. To do so, historical data from a real warehouse is used and processed to create a matrix denoting the flow within each pair of possible goods locations. Then the optimization module is in charge of rearranging the goods based on the shape of the environment to minimize the time spent completing all of the movements registered. The module uses a mathematical warehouse model to calculate the time it takes to move a good between two warehouse locations and evaluate the candidate solutions provided. A Hill–Climbing algorithm with three different neighbourhood operators and a Variable Neighbourhood Search alternating those three operators are implemented and tested in both the real scenario and simplified versions of it.

The rest of the article is structured as follows: The related literature review is presented in Section 2. Next, the warehouse design and restrictions in the proposed optimization scenario are described in Section 3. Next, the proposed optimization methods are presented in Section 4. The experimental results are then presented in Section 5. Finally, Section 6 ends the article by presenting conclusions and future works.

2 Literature review

The design of the layout of packages in a warehouse is a key component and has a significant impact on order picking and movement in the warehouse. It is known that the ideal warehouse layout can be modelled as an optimization problem, where the number of alternatives is very large or infinite, and the evaluation of all alternatives is almost impossible. Therefore, it is considered a complex problem, and approximate algorithms are often used in the solution of this problem. In the design of the warehouse layout, some important factors must be taken into account: the number of blocks; the length, width, number of aisles, shape of the aisles; the number of racking levels; and the position of the entry and exit doors in the warehouse.

The arrangement of packages in a warehouse, i.e. where to place each content, was studied in [6]. Reference is made to different related works up to that year, which had as main objective to find solutions to the internal layout or aisle configuration problem. The literature presented in this work had as a common objective to find the best layout of the warehouse with respect to a given objective function among different layouts that could be adjusted to certain constraints and requirements, and among the most common objective functions, minimizing the travel distance was used.

Jaimes *et al.* [13] proposes a model for redistributing the contents of a warehouse and dividing it into different distinct zones, one zone where orders are prepared and another zone where picking takes place. The model considers random initial storage, which does not take into account considerations about the composition of the products, nor how best to accommodate them, since the case study uses intelligent equipment for the handling and internal transport of the goods, and high shelves are available to store the products. Three costs are proposed to be minimized: those related to the initial investment (construction and maintenance), shortage costs and costs associated with storage policies. While testing the proposed model, they used the layout of a warehouse plant with real data, and shared all the information that can be used to simulate the same scenario.

Mirabelli *et al.* [15] present a model for the allocation of products in a warehouse, the main objective to optimize is to minimize the handling costs that occur in a given period by time slots. They conducted a study in the warehouse of a company that operates in the production and distribution sector of sports equipment and clothing, analysed the volume generated by each product class and the number of manipulations performed in each category. In the solution, they used an algorithm based on the classification of products and obtained a new, more organized and structured warehouse layout, demonstrating that it improves product picking times and delivery times. They conducted experiments on three test cases taking into account the warehouse layout and, for each test case, six synthetically generated scenarios were created while considering a variation of the total amount of products handled. Specifically, decreasing this value by 20%, 10% and 5% and increasing it by the same percentage.

There are different articles that refer to models to design and configure a layout of the packages in a warehouse, and the main objective is to minimize the cost of transportation or distance traveled within the warehouse [9–11, 20]. Moreover, there is some work that focuses on warehouse design to optimize operating costs, minimize congestion in order loading, maximize space utilization, group products by some criteria, etc [16, 21, 22]. Besides, there are works that focus their study on how to optimize the space used, which would mean savings in the use of space and a better distribution of the packages [5, 7].

Ballesteros *et al.* [4] propose the solution to obtain a product distribution in a food warehouse, composed of 3 different areas, 4 material flows and a time horizon of 6 periods. A dynamic allocation of the spaces in the warehouse is performed in the different time periods, in which demand and costs change. To solve the problem they used CPLEX and the results obtained showed that for the instances of the problem the allocation corresponds to the minimum cost of each operational area, which is much more efficient than the distribution they had previously.

Septiani *et al.* [18] proposed a model to obtain an organization of a warehouse, based initially on the collection of information on the expertise of the workers. It is studied with historical data on the input and output of products in the warehouse, the frequency of displacement to find and form order. The proposal improves the distribution of items in the warehouse using a proprietary method that they developed and is based on specific storage. Several test simulations were performed, and the efficiency of the storage space could be calculated. The problem instance used is about a real warehouse structure too specific for the experimentation simulations, divided into six storage areas, each of them with a different polygonal structure.

Arif *et al.* [2] propose a solution in a warehouse that combines several aspects: storage, transmission, distribution and management of stock. The objective of the model is to find solutions to minimize the amount of coverage area used by the pallets and maximize the number of boxes stored on the pallets. For the solution of the problem they use the genetic algorithm and the differential evolution algorithm. In the same year, Sudiarta *et al.* [19] presented an analysis is made of the methods used for planning the distribution of warehouses containing raw materials. The following methods are studied: Dedicated Storage, Class Based Storage, Shared Storage, Random Storage and Herringbone Layout. These methods are used to decrease the material handling distance so that the distribution reaches an optimal solution.

Derhami *et al.* [8] propose a solution for distributing the block stacking of materials, which includes determining the number of aisles and cross aisles, the depth of the lanes and the types of cross aisles. With the results obtained, they show that the resulting layout can save up to 10 % of a warehouse's operating costs. The model is approached as a multi-objective problem, with two objectives to be optimized: maximize space utilization and minimize transportation costs. They created three synthetic instances of the problem with different warehouse sizes, with a minimum and maximum number of aisles to be located, and the results were compared with a real case study on the design of a warehouse for bottled beverages, which bases its storage mainly on block stacking. In the same year, Irman *et al.* [12] proposes a model to obtain an optimal design of the distribution of a warehouse of materials, with the objective of minimizing the costs of travel times within the warehouse. In the solution they use Linear Programming and LINGO 17 software to solve the model.

Saderova [17], presented a methodology for the design of a warehouse system based on 5 phases, which are described by means of a well-detailed flow chart. The results obtained are being applied in a real warehouse application and are using the three proposed alternatives: arrangement of the rack rows, arrangement of the rack blocks and a multi-objective variant with the two previous alternatives. The tests were performed on two different synthetic storage layouts, providing details of the dimensions of each area used and the total number of aisles per area.

Kovács *et al.* [14] present a detailed procedure of the main optimization processes that exist in the literature for warehouse layout design. They comment that there are no uniform and standard procedures for the design, neither in the literature nor in practice, that most of them are unique procedures, and that new works on this subject can contribute significantly to the research. Many of the researches presented only apply the objective function of maximum storage capacity for the selection of the ideal distribution; however, there are other interesting objectives that should be taken into account, such as: maximum utilization of the warehouse area, minimum distances of material flow, minimum investment cost, minimum operating cost, among others. In addition, they present the most common constraints that exist and that must be solved in the optimization models.

After reviewing the literature we have been able to find different models and solutions, there are specific or general contributions, ranging from developments of novel algorithms and better performance to their application to complex and real domains. Furthermore, there are some research papers that present synthetic test instances that could be used as a reference to validate the performance of the proposed solution algorithms for this type of problem. We have been able to see the importance of this problem, and how its complexity has increased due to the variety of factors that have been arising with the need to improve the layout of packages in a warehouse. This paper proposes a use case of a well-known optimization method for a specific warehouse design, which opens the door to future developments in this field.

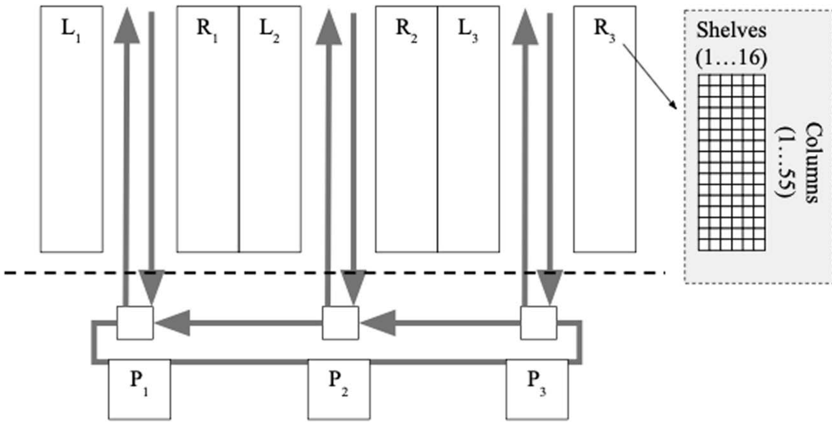

FIGURE 1. Layout of the warehouse, composed by three aisles and three packaging posts.

3 Warehouse layout optimization model

This research is based on data from a real-world automated warehouse. Figure 1 depicts the warehouse layout, which is divided into two sections: a corridor area where stock is stored ($R_{1,2,3}$ and $L_{1,2,3}$) and three packaging posts ($P_{1,2,3}$), where shipments are stored in boxes according to the received shipment order. Each side of the three corridors is divided into 16 shelves and 55 columns, denoted by R or L to indicate right- or left-handedness, respectively. This gives the products a total of 5280 possible locations (3 aisles, 2 sides, 16 shelves and 55 columns). Each one of the possible locations is identified as presented in Equation 1, where a, s and c denote, respectively, the aisle, shelve and column, while the R and L indicate if it is a right- or left-handedness. Packaging locations will also be assumed to have shelf and column values of zero, so they are denoted as $P_{a=\{1,2,3\}}$.

$$\{R \mid L\}_{(a,s,c)},\ \forall a = \{1, 2, 3\},\ \forall s = \{1 \ldots 16\},\ \forall c = \{1 \ldots 55\} \quad (1)$$

In each aisle, an automatic platform transports bins from their respective shelves to the corresponding packaging point and vice versa, or between locations within the same shelf. Each platform can only hold one bin at a time. The three packaging locations are connected by a single unidirectional conveyor belt, which continuously circles from the corridors to the packaging posts ($P_1 \leftarrow P_2 \leftarrow P_3 \leftarrow P_1 \ldots$). The platform is able to directly move a bin between two locations in the same aisle and side, but, in other cases, the bin has to be moved to the corresponding packaging post and then to its final location (in the same or different aisle).

The optimization model will consider minimizing the total time required for operations of movements between locations within the warehouse, so the following assumptions will be used to calculate this time:

- At speeds of S_h and S_v, each of the robotic platforms moves both horizontally (between columns) and vertically (between shelves) at the same time.
- Distances d_h and d_v denote the distances between consecutive columns or shelves, respectively.
- To move a bin from one side of an aisle to the other ($R_{(a,s1,c1)}$ to $L_{(a,s2,c2)}$ or vice versa), the platform moves it from the origin to the corresponding packaging point (P_a), then to the destination.

- The platform will move a bin between aisles by moving it to the corresponding packaging point, then the conveyor belt will move it to the destination packaging point, and finally to the destination. The distance between packaging points is denoted as d_p, and the belt's outer trail is d_r. Bins are moved along the conveyor belt at a speed of S_b.

The computation algorithm has been formulated according to the warehouse's layout, while the speed and distance values are the ones provided by the warehouse owners. The technical specifications of the automated platforms have been considered to establish the values of the variables, which are as follows: $S_h = 2.5$, $S_v = 1.5$ and $S_b = 2.5$ for speeds (in m/s), as well as $d_h = 2.5$, $d_v = 1$, $d_p = 2$, $d_r = 10$, in meters, for distances in the layout. Given so, the time needed of moving a bin from an origin ($O_{a1,s1,c1}$) to a destination ($D_{a2,s2,c2}$) is calculated as presented in Algorithm 1.

Algorithm 1: Calculation of the time for moving a bin among two points.

1 **Input**: $a1$, $a2 \rightarrow$ aisle (1, 2 or 3) of origin and destination locations;
2 **Input**: O, $D \rightarrow$ side (left or right) of origin and destination locations;
3 **Input**: $s1$, $s2 \rightarrow$ shelve (1 to 16) of origin and destination locations;
4 **Input**: $c1$, $c2 \rightarrow$ column (1 to 55) of origin and destination locations;
5 **if** $a1 == a2$ **then**
6   **if** $O == D$ **then**
7   time $= max(\frac{|c1-c2|d_h}{S_h}, \frac{|s1-s2|d_v}{S_v})$
8   **else**
9   time $= max(\frac{(56-c1)d_h}{S_h}, \frac{(17-s1)d_v}{S_v}) + max(\frac{(56-c2)d_h}{S_h}, \frac{(17-s2)d_v}{S_v})$
10 **else**
11   time $= max(\frac{(56-c1)d_h}{S_h}, \frac{(17-s1)d_v}{S_v}) + max(\frac{(56-c2)d_h}{S_h}, \frac{(17-s2)d_v}{S_v})$;
12   **if** $a1 > a2$ **then**
13   time $= time + \frac{(a1-a2)d_r}{S_b}$
14   **if** $a1 < a2$ **then**
15   time $= time + \frac{(a1-1)d_p+(3-a2)d_p+d_r}{S_b}$
16 $return(time)$;

The algorithm considers four different types of movements, which are illustrated in Figure 2 and explained next.

1. Case (a): Since the origin and destination are on the same aisle and side, and the bin is moved directly by the robotic platform.
2. Case (b): As both locations are on opposite sides of the same aisle, the bin must first be moved to the packaging point, then to the final destination.
3. Case (c): Destination is in an aisle in the direction of the conveyor belt, so the time of movements between packaging posts has to be considered.
4. Case (d): The destination is in an aisle in the opposite direction of the conveyor belt, so the bin must circulate along the outer ring.

The calculation of time for each one of the four cases is done as indicated in Algorithm 1 at lines 7, 9, 13 and 15, respectively.

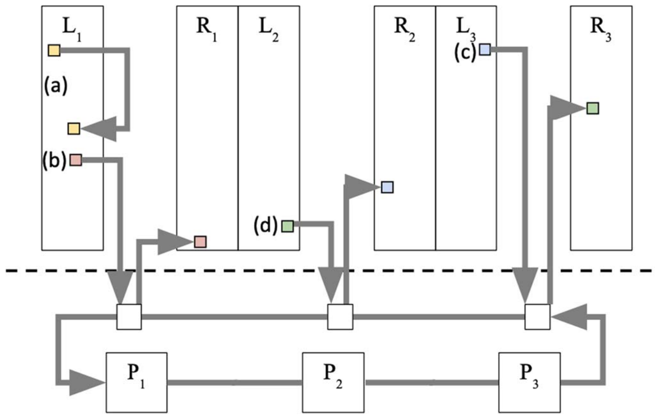

FIGURE 2. Visual representation of the possible movements. With origin and destination in: same aisle and side (a), same aisle but different side (b), different aisle without using the external ring (c) or using it (d).

The optimization goal is the reduction of the total time spent in the internal movements of bins within the warehouse, by reducing the time it takes to retrieve and move them from their shelves to the conveyor belt. To do so, optimization methods will be used to reduce the amount of time that robotic platforms spend moving bins by relocating them throughout the warehouse.

The fitness function to minimize is presented in Equation 2 to formulate the optimization algorithm, where both *Dist* and *Flow* are 5280×5280 matrices denoting the distance and number of movements realized (flow) between two locations, respectively.

$$fitness = \sum_{i,j \in 1...5280} (Dist_{i,j} \cdot Flow_{i,j}) \tag{2}$$

Over each pair of locations, *Dist* is calculated using the procedure presented in Algorithm 1, whereas real data from a company is used to obtain the values of *Flow*. With a total of 373933 entries, the data covers the entire historical information of warehouse movements from 2019-09-26 to 2020-09-15. The data is processed, and the number of movements between each pair of locations is calculated, in order to fill *Flow*.

4 Optimization procedure

The optimization procedure will be in charge of minimizing Equation 2. For this purpose, a permutation encoding is used, thus, a candidate bin arrangement is represented as $X = (x_1, x_2, ...x_{5280})$, where the value x_i denotes the identifier of the product stored in the i-th location of the warehouse.

The initialization of the solution to be improved will be done as $x_i = i$, indicating that the initial solution is the warehouse's original layout. This ensures that the neighbour solutions are close to the original arrangement, saving on the cost of rearranging all of the elements within the warehouse.

Different switching operators are used in this work to generate neighbouring solutions and explore the solution space. Any change in the arrangement of the positions will result in a reordering of both the rows and columns in *Flow*, requiring a recalculation of the result of Equation 2.

In this work, three alternatives for obtaining neighbour solutions are implemented, which are the following:

1. *swap*: This operator takes randomly two indexes of the solution and then interchanges their values.
2. *insertion*: In order to insert one of the solution's elements into a random location in the vector, the operator selects one of the solution's elements and a random location in the vector to insert it.
3. *adjacent*: This operator chooses one of the warehouse elements at random and swaps it with one of the colliding ones (upper or lower shelve or column in the left or right).

Two different optimization procedures are tested using those mutation operators, which are versions of the well-known Hill–Climbing (HC) and Variable Neighbourhood Search (VNS) algorithms. Both algorithms are presented respectively in Algorithms 2 and 3. It is worth mentioning the main difference in terms of the computational cost relies on the mechanism for modifying the selected operator implemented by VNS (3 at line 16), so the algorithmic complexity of both alternatives remains similar.

Algorithm 2: Hill-Climbing optimization procedure.

```
Inputs: op, N_s ;
solution = {1, 2, 3, ... 5280};
time_best = fitness(solution);
for i ∈ {1 ... iterations} do
    time_neighbour = ∞
    for j ∈ {1 ... N_s} do
        newSolution = getNeighbour(solution, op);
        time_new = fitness(newSolution);
        if time_new < time_neighbour then
            time_neighbour = time_new;
            neighbour = new;
    if time_neighbour < time_best then
        time_best = time_neighbour;
        solution = neighbour;
return(solution);
```

In both cases, following the generation and evaluation of the initial solution, the methods iteratively generate N_s alternative plans derived from variations of the initial solution according to the selected operators (*op*). After all of the neighbours solutions have been generated, the one with the lowest time replaces the original only if it is improved.

5 Experimentation and results

This section presents the results of the experiments conducted to compare the algorithms' performance, as well as for different sizes of the explored neighbourhood of a solution (N_s). In order to provide a fair comparison between the methods in terms of the number of evaluations of the fitness function (Eq. 2), the number of iterations the algorithm is ran is established at *iterations* $= 10^5/N_s$. The results are shown in Table 1, where the different executions are compared in terms of execution time and fitness function reduction value compared to the initial (original) bin arrangement.

When comparing the Hill–Climbing algorithms alone, insertion and swapping get similar results in terms of the overall achieved fitness during the experiments, although their execution times do differ a bit, swapping being slightly faster than insertion. The adjacent algorithm is faster than these two, but it is also less efficient across the table, only achieving a fitness value lower than 95% in a single experiment. Even if both, insertion and swapping have yielded similar results, among all the experiments, swapping is the one that has reduced the fitness value the most by using the Hill–Climbing algorithm.

Algorithm 3: Variable Neighbourhood Search optimization procedure.

```
Inputs: op, N_s ;
op = swap;
solution = {1, 2, 3, ... 5280};
time_best = fitness(solution);
for i ∈ {1 ... iterations} do
    time_neighbour = ∞
    for j ∈ {1 ... N_s} do
        newSolution = getNeighbour(solution, op);
        time_new = fitness(newSolution);
        if time_new < time_neighbour then
            time_neighbour = time_new;
            neighbour = new;
    if time_neighbour < time_best then
        time_best = time_neighbour;
        solution = neighbour;
    else
        if op == swap then
            op = insertion
        else
            if op == insertion then
                op = adjacent
            else
                op = swap
return(solution);
```

Thanks to the fact that, as explained earlier, VNS uses the three aforementioned algorithms, it has been able to obtain the best fitness value by a wide margin, at 80.89%, in spite of performing the worst of all in terms of execution time, doubling in length some of the slowest attempts. All the other executions of the VNS algorithm were much faster, and still of similar efficiency to its Hill–Climbing counterparts, adjacent being the sole exception to this.

TABLE 1. Experimental results obtained. The result is calculated as the percent time reduction over the initial arrangement.

Algorithm (N_s)	**iterations**	**% fitness**	**Time**
HC_{swap} ($N_s = 10$)	10000	99.93	3:48
HC_{swap} ($N_s = 50$)	2000	**89.40**	6:16
HC_{swap} ($N_s = 100$)	1000	93.23	5:52
HC_{swap} ($N_s = 250$)	400	98.88	4:00
HC_{swap} ($N_s = 500$)	200	98.90	3:22
HC_{swap} ($N_s = 1000$)	100	99.57	3:25
$HC_{insertion}$ ($N_s = 10$)	10000	96.18	5:43
$HC_{insertion}$ ($N_s = 50$)	2000	**92.81**	6:12
$HC_{insertion}$ ($N_s = 100$)	1000	94.13	4:17
$HC_{insertion}$ ($N_s = 250$)	400	97.11	4:54
$HC_{insertion}$ ($N_s = 500$)	200	99.39	3:02
$HC_{insertion}$ ($N_s = 1000$)	100	99.19	3:17
$HC_{adjacent}$ ($N_s = 10$)	10000	100	3:52
$HC_{adjacent}$ ($N_s = 50$)	2000	98.86	5:06
$HC_{adjacent}$ ($N_s = 100$)	1000	**94.51**	4:28
$HC_{adjacent}$ ($N_s = 250$)	400	98.82	4:41
$HC_{adjacent}$ ($N_s = 500$)	200	99.20	3:21
$HC_{adjacent}$ ($N_s = 1000$)	100	99.33	3:06
VNS ($N_s = 10$)	10000	**80.89**	14:07
VNS ($N_s = 50$)	2000	93.68	6:07
VNS ($N_s = 100$)	1000	96.53	4:30
VNS ($N_s = 250$)	400	98.80	4:51
VNS ($N_s = 500$)	200	99.32	3:18
VNS ($N_s = 1000$)	100	99.72	3:25

It is also worth noting that the top result of each algorithm always has N_s equal to or lower than 100, as well as the one that takes the most amount of time to complete, which indicates that it is preferred to keep the neighbourhood small and perform a high number of iterations in exchange for a higher execution time. VNS with $N_s = 10$ achieves the lowest fitness value for the optimization of the warehouse, a value that obtained some of the worst results when used with the Hill–Climbing algorithms.

6 Conclusions and future works

This work has presented the modelling of an optimization problem from a real company's historical inventory management data. The characteristics of the automatic platforms that operate the warehouse and their rates of movement are used to create a mathematical model for the operation of the warehouse. Additionally, historical data on past bonding movements within the area are used to model the matrix containing the flow or number of movements between sites in the warehouse.

An optimization procedure is defined to properly arrange goods in the warehouse to improve efficiency and reduce costs. The proposed solution uses the Variable Neighbourhood Search algorithm to find the most efficient arrangement of products within the warehouse. This algorithm is compared to the Hill–Climbing with different operators, but none of them improves the performance of the VNS, which is measured in terms of both computation time and reduction in operation time. However, even if the VNS requires more computational resources, the operation time is reduced considerably enough to accept this algorithm as the best one overall.

Optimizing the layout of a warehouse can have a significant impact on the efficiency of operations. For instance, by reducing delivery preparation time: By organizing the warehouse layout to minimize the time materials need to travel. On the other hand, it can reduce bottlenecks and delays. Overall, properly optimizing a warehouse layout can have a wide range of benefits for the efficiency and effectiveness of operations.

Future work will focus on real-time warehouse management, including the optimization process within warehouse operations to ensure optimal allocation of goods when they first arrive. Additionally, exploring several algorithm variants to find a configuration that elevates the optimization algorithm to a higher sophistication level. On the other hand, other approaches, such as evolutionary algorithms or ant colony approaches, will be considered to obtain better results.

Acknowledgements

The work was supported in part by the European Union's Horizon 2020 Research and Innovation Programmes under Grant 861540, and in part by the Spanish Ministry of Science and Innovation through a research project under Grant PID2019-109393RA-I00.

References

[1] F. N. Arif, B. Irawan and C. Setianingsih. Optimization of storage allocation for final goods warehouse using genetic algorithm. In *2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pp. 1–6, 2020.

[2] F. N. Arif, B. Irawan and C. Setianingsih. Optimization of storage allocation for final goods warehouse using genetic algorithm. In *ICECIE 2020—2020 2nd International Conference on Electrical, Control and Instrumentation Engineering, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2020.

[3] P. Baker and M. Canessa. Warehouse design: a structured approach. *European Journal of Operational Research*, **193**, 425–436, 2009.

[4] F. A. Ballesteros-Riveros, M. D. Arango-Serna, W. Adarme-Jaimes, J. A. Zapata-Cortes, F. A. Ballesteros-Riveros, M. D. Arango-Serna, W. Adarme-Jaimes and J. A. Zapata-Cortes. Storage allocation optimization model in a Colombian company. *DYNA*, **86**, 255–260, 2019.

[5] L. F. Cardona and K. R. Gue. How to determine slot sizes in a unit-load warehouse. *IISE Transactions*, **51**, 355–367, 2018.

[6] R. de Koster, T. Le-Duc and K. J. Roodbergen. Design and control of warehouse order picking: a literature review. *European Journal of Operational Research*, **182**, 481–501, 2007.

[7] S. Derhami. *Designing Optimal Layouts for Block Stacking Warehouses*. PhD Thesis, 2017.

[8] S. Derhami, J. S. Smith and K. R. Gue. A simulation-based optimization approach to design optimal layouts for block stacking warehouses. *International Journal of Production Economics*, **223**, 107525, 2020.

[9] K. R. Gue and R. D. Meller. A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points. *European Journal of Operational Research*, **236**, 382–394, 2014.

[10] K. R. Gue, G. Ivanović and R. D. Meller. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review*, **48**, 795–806, 2012.
[11] K. R. Gue and R. D. Meller. Aisle configurations for unit-load warehouses. *IISE Transactions*, **41**, 171–182, 2009.
[12] A. Irman, Y. Muharni and A. Yusuf. Design of warehouse model with dedicated policy to minimize total travel costs: a case study in a construction workshop. *IOP Conference Series: Materials Science and Engineering*, **909**, 012088, 2020.
[13] W. A. Jaimes, M. A. Otero, T. A. Pineda, R. Quiñones and L. T. López. Optimization of a warehouse layout used for storage of materials used in ship construction and repair. *Ship Science & Technology*, **5**, 59, 2012.
[14] G. Kovács. Special optimization process for warehouse layout design. In *Lecture Notes in Mechanical Engineering*, vol. 22, pp. 194–205. Springer Science and Business Media Deutschland GmbH, 2021.
[15] G. Mirabelli, T. Pizzuti and D. Lagana. Adaptation of the multi-layer allocation problem for warehouse layout optimization: a case study. In *Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS 2013*, vol. 1 pp. 167–172, 2013.
[16] M. Moshref-Javadi and M. R. Lehto. Material handling improvement in warehouses by parts clustering. *International Journal of Production Research*, **54**, 4256–4271, 2016.
[17] J. Saderova, A. Rosova, M. Sofranko and P. Kacmary. Example of warehouse system design based on the principle of logistics. *Sustainability*, **13**, 4492, 2021.
[18] W. Septiani, G. A. Divia and S. Adisuwiryo. Warehouse layout designing of cable manufacturing company using dedicated storage and simulation Promodel. *IOP Conference Series: Materials Science and Engineering*, **847**, 012054, 2020.
[19] N. Sudiarta, L. Gozali, I. A. Marie, I. Wayan and L. T. Sukania. Comparison study about warehouse layout from some paper case studies. *IOP Conference Series: Materials Science and Engineering*, **852**, 012112, 2020.
[20] L. M. Thomas and R. D. Meller. Analytical models for warehouse configuration. *IIE Transactions*, **46**, 928–947, 2014.
[21] L. M. Thomas and R. D. Meller. Developing design guidelines for a case-picking warehouse. *International Journal of Production Economics, Part C*, **170**, 741–762, 2015.
[22] S. Zhou, Y. Gong and R. De Koster. Designing self-storage warehouses with customer choice. *International Journal of Production Research*, **54**, 3080–3104, 2016.

Received 20 May 2022