Optimization of Warehouse Layout for the minimization of operation times

Jon Díaz, Haizea Rodriguez, Jenny Fajardo-Calderín, Ignacio Angulo, and Enrique Onieva

DeustoTech, Faculty of Engineering, University of Deusto, Av. Universidades, 24, 48007 Bilbao, Spain
{jondi,haizearodriguez}@opendeusto.es, {fajardo.jenny, ignacio.angulo, enrique.onieva}@deusto.es

Abstract. Warehousing management is essential for many companies involved in the supply chain. The optimal arrangement and operation of warehouses play important role in companies, allowing them to maintain and increase their competitiveness. One of the main goals of warehousing is the reduction of costs and improvement of efficiency.
Determination of the ideal warehouse layout is a special optimization case. In the article, a procedure is applied to the optimization of the allocation of bins in an automated warehouse with special characteristics. The initial layout of the warehouse, as well as the automated platforms constraints the search, as well as define the time needed to move goods inside the warehouse. The definition of time needed to move goods, with the analysis of historical data, allows the definition of a mathematical model of the operation of the warehouse. Using that model, an optimization procedure based on the well-known hill-climbing algorithm is defined. Experimental results show increments in the efficiency of the warehousing operations.

Keywords: Warehouse layout design · Optimization of warehouse · Decision support models · Logistics · Optimization

1 Introduction

One of the principal goals of global trends of the market is to put a great effort into trying to distribute goods more rapidly to the customers and consequently reduce the cost of item storage. An initial idea of directly connecting the supplier and the customer has been contemplated, but it is still not a reality [3]. Companies are still in need of a warehouse where goods can be held and organized before distribution in a clever way, and intelligent warehouses are the key.

Storehouses have evolved over the years. At first, all the manhandling of the products was done manually, but in the effort of reducing time and cost, technology has been used to automate the main processes. The machinery installed in the warehouses has proved useful by improving the movement's cost and efficiency. Nonetheless, there are some obstacles that even advanced machines cannot solve. One of the main obstacles found at warehouses, while trying to optimize the services provided, is the proper arrangement of goods [1].

Warehouses use to have a static layout that is not supposed to change, but the way products are arranged can be adapted to the overall needs. The main goal is the reduction of time, and placing the most requested items closer to the extraction points can help achieve it. These products are constantly being moved around, thus, by reducing the effort to access them, the overall cost is then reduced. The effect of this is similarly reflected in both, manually managed warehouses (and) automated ones.

In order to find the good's best arrangement, the use of optimization techniques is crucial. The solution provided by the algorithms can accomplish the goal of reducing the cost and improving the overall performance, but it cannot assure a perfect solution is found. Because of this, several algorithms are tested with the intention of finding the arrangement that returns the best warehouse layout.

This article presents a use case of optimization methods applied to the arrangement of the goods within an automated warehouse. The objective of the optimization procedure is to minimize the time the automated platforms inside the warehouse spend moving goods. To achieve so, historical data from a real warehouse is used and processed to build a matrix denoting the flow within each pair of possible locations of goods and then, using the shape of the environment, the optimization module is in charge of re-arranging the goods to minimize the time spent to realize all the movements registered. The module used a model of the warehouse to model the time needed to move a good between two locations of the warehouse. A Hill-Climbing algorithm is implemented and 3 different neighbourhood operators are compared in terms of execution time and performance obtained. This paper is structured as follows. Section 2 presents the related literature review. Section 3 describes the warehouse design and restrictions applicable in the proposed scenario. Section 4 states the formulation of the problem to be solved by the optimization. After that, Section 5 presents experimental results obtained. Finally, Section 6 states conclusions and future works.

2 Literature review

The selection of the ideal warehouse layout is a special optimization process, not a typical mathematical optimization. Since the number of layout alternatives is huge or infinite, therefore, the formation and the evaluation of all possible alternatives are impossible. This is one of the reasons that heuristic method and continuous iteration have to be used during the optimization process.

It is a complex problem to design a warehouse. It includes a large number of interrelated decisions involving the functional description of a warehouse, technical specifications, selection of technical equipment and their layout, warehouses processes, warehouses organizations and others. Several authors present general models of warehouse design and planning.

Jaimes et al. [7] presented the evaluation of applying models to improve efficiency in management of warehouses used in shipyards, focused on pick up, packing, and shipping activities. Besides proposing the best physical layout for the storage of goods, the model seeks to minimize three types of costs: costs related to the initial investment (construction and maintenance), shortage costs, and costs associated with storage policies.

Muharni et al. [10] presented an optimization model to obtain a better disposition of a warehouse containing raw materials. For the solution to the design of the arrangement of the elements, the particle swarm algorithm was used and the objective was to minimize the cost of handling materials. Moreover, Ballesteros et al. [4] propose an optimization model to obtain an adequate arrangement of different food products in a warehouse, over different periods of time. They develop a solution that identifies operational areas and the amount of required spaces in which the products must be located, seeking to reduce the costs of maintaining and handling the products. The CPLEX software is used to solve the problem of dynamic allocation of products.

Arif et al. [2] proposed the Genetic Algorithm application in optimizing the arrangement of storage of manufactured goods in warehouses. The goal is to find solutions to minimize the amount of coverage area used by pallets and maximize the number of boxes stored on pallets. The use of Genetic Algorithms in this research aims to find the best fitness value in the allocation of goods in the finished warehouse so that the arrangement and allocation of goods are not done carelessly and can reduce the free space in the warehouse.

Derhami et al [5] presented an optimal design of the distribution of a warehouse, taking into account: the number of aisles and cross aisles, the depth of the streets and the types of cross aisles. They develop a simulation-based optimization algorithm to find optimal arrangements regarding: material handling cost and space utilization. They use a case study in the beverage industry and show that the resulting layout can save up to 10% of a warehouse's operating costs. Also, Irman et al. [6] developed an integer linear programming model to obtain an optimal layout design that minimizes total travel costs, and they also used the LINGO 17 software.

Sudiarta et al. [11] In this work an analysis of the methods used for the planning of the warehouse distribution is made, they study the following methods: Dedicated Storage, Class-Based Storage, Shared Storage, Random Storage and Fishbone Layout. These methods are used to decrease the handling distance of the materials, so that the distribution reaches an optimal solution.

Kovács et al. [8] presented the characteristics and the detailed procedure of the special optimization process of the warehouse layout design are described. They describe the most common objective functions that can be taken into account to model the problem. In addition, they explain the most important constraints and limitations that the optimization problem has.

Saderova [9] presented warehouse system design methodology that was designed applying the logistics principle-systematic (system) approach. The starting point for designing a warehouse system represents of the process of design logistics systems.

Within literature presented, different specific use cases and problem modeling can be found, with contributions ranging from developments of novel and better performing algorithms to their application to real complex domains. This works is focuses on the application of optimization procedures to a real case, with specific modelling, requirements and constraints, which makes it interesting for the community. The present article proposes an use case of well-known optimization method to specific warehouse layout, which opens the door to future developments in the field.

3 Characteristics of the warehouse layout design

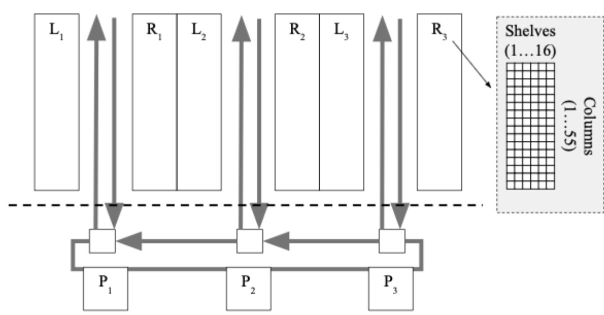

Fig. 1. Layout of the warehouse, composed by three aisles and three packaging posts.

This work is done using data from a real automated warehouse. An schema of the layout of the warehouse can be seen in Figure 1, which is divided in two sections, the corridor area where the stock is stored ($R_{1,2,3}$ and $L_{1,2,3}$) and three packaging posts ($P_{1,2,3}$), where the shipment is stored in boxes according to the received shipment order. The three corridors are divided into sides, denoted by R or L to denote right of left handed respectively, and each side is divided into 16 shelves and 55 columns.

This makes a total number of 5280 possible locations for the products (3 aisles · 2 sides · 16 shelves · 55 columns). Each location will be identified by variable $R_{\{a,s,c\}}$, $\forall a = 1, 2, 3$, $\forall s = 1 \ldots 16$, $\forall c = 1 \ldots 5$ (similarly for $L_{\{a,s,c\}}$ for left handed sides). In addition, packaging locations will assumed to have shelf and column equal to zero.

A single unidirectional conveyor belt connects both sections of the warehouse, which continuously circles from the corridors to the working posts. An automatic platform operates in each of the aisles, taking the bins from their respective shelves to the belt and vice versa. The platform can only take a single bin at a time, and the belt will completely stop its movement until the bin has either been collected or placed on top of the belt itself.

The platform is also able of interchanging the position of two bins of its corridor. When the bins are on the same side, swapping is almost immediate, but, if that is not the case, then the bin needs to be placed on the belt so that it can be properly placed once it comes back. Therefore, swapping bins of different sides is extremely expensive.

Each of the aisles is equipped with a robotic platform able to pick a bin from a particular location and to move to another in the same aisle, or to put it in the conveyor belt, in order to move it to another one or to one of the three packaging posts. The optimization will consider the minimization of the total time needed for operations of movements between locations inside the warehouse, so in order to calculate this time, following assumptions are considered:

- Each of the robotic platforms moves both, horizontally (between columns) and vertically (between shelves) simultaneously, at speeds S_h and S_v respectively.
- Distances between consecutive columns or shelves are denoted as d_h and d_v, respectively.
- In order to move a bin between two sides of the same aisle (R_i to L_i or vice versa), the platform will first move the bin from the origin to the corresponding packaging point (P_i), and then to the destination.
- To move a bin between different aisles, the platform will move it to the corresponding packaging point, then, the conveyor belt moves it to the destination packaging point and then to the destination. Distance between packaging points is denoted as d_p, while the outer trail of the belt is a total of d_r. The conveyor belt moves bins at speed S_b.

4 Mathematical Model

Given characteristics exposed in previous section, the time needed of moving a bin from a origin ($O_{a1,s1,c1}$) to a destination ($D_{a2,s2,c2}$) is calculated as presented in Algorithm 1.

In this case, it has been considered $S_h = 2.5$, $S_v = 1.5$ 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. The algorithm takes into account four different cases of movements, which are visually explained in Figure 2:

1. Origin and destination are in the same aisle and side: the robotic platform directly moves the bin. Case (a) in Figure 2.
2. If both locations are in opposite sides of the same aisle, the bin must be moved to the corresponding packaging point, and then to the destination. This is represented in Case (b) in Figure 2.
3. If origin and destination are in different aisles, the time will depend if the movement needs to make use of the outer ring of the conveyor belt or not.

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

if $a1 == a2$ **then**
  if $O == D$ **then**
    time $= max(\frac{|c1-c2|d_h}{S_h}, \frac{|s1-s2|d_v}{S_v})$
  else
    time $= max(\frac{(56-c1)d_h}{S_h}, \frac{(17-s1)d_v}{S_v}) + max(\frac{(56-c2)d_h}{S_h}, \frac{(17-s2)d_v}{S_v})$
else
  time $= max(\frac{(56-c1)d_h}{S_h}, \frac{(17-s1)d_v}{S_v}) + max(\frac{(56-c2)d_h}{S_h}, \frac{(17-s2)d_v}{S_v})$;
  if $a1 > a2$ **then**
    time $= time + \frac{(a1-a2)d_r}{S_b}$
  if $a1 < a2$ **then**
    time $= time + \frac{(a1-1)d_p+(3-a2)d_p+d_r}{S_b}$

The first case is represented in case (d) in Figure 2, while the second in case (c).

The main goal is to accelerate the movement of the bins across the warehouse by decreasing the time spent retrieving and moving them from their respective shelves to the belt. For doing so, optimization methods will be used to minimize the time spent by robotic platforms moving bins by reallocating them along the warehouse.

In order to formulate the optimization algorithm, the fitness function to minimize is presented in Equation 1, where both M_d and M_f are 5280×5280 matrices denoting the distance and number of movements realized (flow) between 2 locations respectively.

$$F = \sum(M_d \times M_f) \tag{1}$$

M_d is calculated by using the procedure presented in Algorithm 1 over each pair of locations, while, for obtaining the values of M_f, real data coming from a company is used. Data covers the entire historical information of movements in the warehouse from 2019-09-26 to 2020-09-15, having a total of 373933 entries. In order to fill M_f, the data is processed, and the number of movements between each pair of locations is calculated.

4.1 Optimization procedure

For the minimization of the Equation 1, a permutation based coding is used. A candidate solution represents a feasible arrangement of the bins, which can be represented as $X = (x_1, x_2, ...x_{5280})$, where x_i represents the product stored in the i-th location.

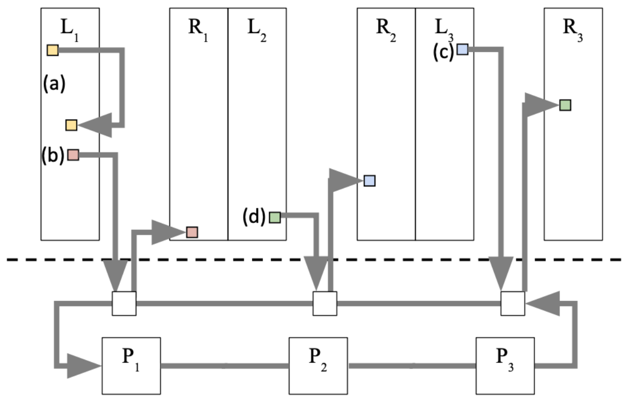

Fig. 2. Visual representation of the possible movements. With origin and destination in: same aisle and side (a), same aisle but different side (b), different aisle without using the external ring (c) or using it (d).

Initially, $x_i = i$, denoting the initial solution being the original arrangement of the warehouse. With this, we ensure the final solution will be as similar as the original arrangement as possible, avoiding the cost of arranging all the elements within the warehouse from scratch. Interchanging operators will allow to generate neighbour solutions to explore the solutions space. Any change in the arrangement of the positions will derive in a reordering of both the rows and columns in M_f, so result of Equation 1 will be recalculated.

The optimization procedure can be seen in Algorithm 2, a version of the well known Hill-Climber algorithm, where the number of neighbour solutions to explore in each iteration is limited, due the complexity of the solution space. The selection of the Hill-Climber schema is justified by the need of the stakeholder of the solution for using an algorithm able to modify as less as possible the initial arrangement of the warehouse, while guaranteeing as optimal as possible results.

After the initial plan is generated and evaluated using the procedure presented in Algorithm 1, the method, in an iterative way, creates N_s alternative plans derived from variations of the initial solution, and the time of each one is subsequently calculated and stored. Once all the alternatives have been evaluated, the solution with the lowest cost replaces the initial one, but only if an improvement is made.

In this work, three alternatives for the *getNeighbour* operator (line 6 in Algorithm 2) are implemented, which are following:

1. *Swap*: This operator takes randomly two elements of the solution and interchanges them.

Algorithm 2: Optimization procedure.

```
solution = {1, 2, 3, ... 5280};
time = timeCalculation(solution);
for i ∈ {1 ... iterations} do
    bestTime = ∞
    for j ∈ {1 ... N_s} do
        newSolution = getNeighbour(solution);
        newTime = timeCalculation(newSolution);
        if newTime < bestTime then
            bestTime = newTime;
            besSolution = newSolution;
    if bestTime < time then
        time = bestTime;
        solution = bestSolution;
return(solution);
```

2. *Insertion*: The operator selects one of the elements of the solution and a random location in the vector in order to insert it in.
3. *Adjacent*: The operator randomly selects one of the elements in the warehouse and interchanges it with one of the colliding ones (upper or lower shelve or column in the left or right).

5 Experimentation and results

This section presents the results of the carried out experiments for comparison of the performance of the algorithm using the three proposed neighbour operators, as well as for different sizes of the explored neighbour of a solution (N_s). During the experimentation, the number of iterations was kept constant at $iterations = 10^5$. Results are presented in Table 1, where the different executions ran are compared in terms of execution time and reduction value of the fitness function with respect to the initial (original) arrangement of the bins.

The *Insertion* algorithm was discarded as it took more than 5 hours to perform the 10.000 iterations with the lowest N_s. Since in subsequent executions the neighborhood size was greatly increased, it was expected for the second algorithm to take much longer, which was proven to be true with the neighborhood size of 500. This happened due to the actual implementation of the operator did not allow the use of fast calculation of the fitness function, which was achieved by subtracting and adding values to the previous fitness accordingly to movements performed.

As far as the rest of the algorithms are concerned, their execution times are very close to each other. However, the gap among them increases as the neighborhood becomes bigger.

getNeighbour()	N_s	Execution Time	Result
Swap	100	00:27:50	1.49%
Swap	500	01:24:13	4.15%
Swap	1000	03:16:24	7.69%
Insertion	100	05:27:21	10.19%
Insertion	500	26:24:06	64.98%
Insertion	1000	-	-
Adjacent	100	00:29:59	22.46%
Adjacent	500	01:40:56	54.39%
Adjacent	1000	03:10:39	68.90%

Table 1. Experimental results obtained. Result is calculated as the percent time reduction over the initial arrangement.

Based on the result of the experimentation procedure, it can be concluded that the first algorithm does not achieve considerable improvements with the enlargement of the neighborhood. The third algorithm performs better, and it reaches a good optimization values when compared to the other methods while maintaining reasonable amount of time spent.

6 Conclusions and future works

This work has presented the modelization of an optimization problem from the historical data of warehousing operations in a real company. The characteristics of the automated platforms operating the warehouse, as well as their movement speeds are used to build a mathematical model of the operation of the warehouse. In addition, historical data regarding past movements of bind within the area are used to model the matrix containing the flow or number of movements between locations in the warehouse.

An optimization procedure is defined to properly arrange the goods in the warehouse for improved efficiency and reduction of costs. The proposal is based on the well-known hill-climbing algorithm, and it is used for the arrangement of products within the warehouse. Three different neighbouring operators are used within the procedure and the performance is measured in terms of both computational time and reduction of operational time.

Future works will be focused on the real-time management of the warehouse, including the optimization procedure within the operation of the warehouse to guarantee the optimal allocation of goods as they arrive for the first time. The exploration of additional operators for modification of the current arrangement, as well as the implementation of more sophisticated optimization algorithms, will be considered in future developments. With this regard, evolutive meta-heuristics able to manage with different recombination or modification methods will be considered as the next step to improve actual results.

Acknowledgements

This work has been supported by The LOGISTAR project, which has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 769142. Also, from the Spanish Government under the project IoTrain (RTI2018-095499-B-C33).

References

1. Arif, F.N., Irawan, B., Setianingsih, C.: 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 (Nov 2020). https://doi.org/10.1109/ICECIE50279.2020.9309669
2. Arif, F.N., Irawan, B., Setianingsih, C.: 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. (nov 2020). https://doi.org/10.1109/ICECIE50279.2020.9309669
3. Baker, P., Canessa, M.: Warehouse design: A structured approach. European Journal of Operational Research **193**(2), 425–436 (2009). https://doi.org/https://doi.org/10.1016/j.ejor.2007.11.045
4. Ballesteros-Riveros, F.A., Arango-Serna, M.D., Adarme-Jaimes, W., Zapata-Cortes, J.A., Ballesteros-Riveros, F.A., Arango-Serna, M.D., Adarme-Jaimes, W., Zapata-Cortes, J.A.: Storage allocation optimization model in a Colombian company. DYNA **86**(209), 255–260 (apr 2019). https://doi.org/10.15446/DYNA.V86N209.77527
5. Derhami, S., Smith, J.S., Gue, K.R.: A simulation-based optimization approach to design optimal layouts for block stacking warehouses. International Journal of Production Economics **223**, 107525 (may 2020). https://doi.org/10.1016/J.IJPE.2019.107525
6. Irman, A., Muharni, Y., Yusuf, A.: 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**(1), 012088 (dec 2020). https://doi.org/10.1088/1757-899X/909/1/012088
7. Jaimes, W.A., Otero Pineda, M.A., Quiñones, T.A.R., Tejeda López, L.: Optimization of a warehouse layout used for storage of materials used in ship construction and repair. Ship Science & Technology **5**(10) (2012)
8. Kovács, G.: 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 (nov 2021)
9. Saderova, J., Rosova, A., Sofranko, M., Kacmary, P.: Example of Warehouse System Design Based on the Principle of Logistics. Sustainability **13**(8), 4492 (apr 2021). https://doi.org/10.3390/su13084492
10. Septiani, W., Divia, G.A., Adisuwiryo, S.: Warehouse Layout Designing of Cable Manufacturing Company using Dedicated Storage and Simulation Promodel. IOP Conference Series: Materials Science and Engineering **847**(1), 012054 (apr 2020). https://doi.org/10.1088/1757-899X/847/1/012054

11. Sudiarta, N., Gozali, L., Marie, I.A., Sukania, I.W.: Comparison Study about Warehouse Layout from Some Paper Case Studies. IOP Conference Series: Materials Science and Engineering **852**(1), 012112 (jul 2020). https://doi.org/10.1088/1757-899X/852/1/012112