A New Layout Problem for Order-Picking Warehouses

Ömer Öztürkoğlu
Department of Business Administration
Yaşar University
Izmir, Turkey
omer.ozturkoglu@yasar.edu.tr

Ayşegül Mağara
Schneider Electric
Izmir, Turkey
aysegul.magara@gmail.com

Abstract

Recent studies of warehouse layout designs show that the travel distance for order-picking operations can be reduced by changing the angle of the cross aisles in the traditional warehouse layout. Thus, in this study, a new design idea is proposed to search for better layouts for order-picking operations than traditional two-block layouts. For this, two angled cross aisles are allowed to intersect in the middle of the storage area. A new constructive aisle model is developed so as to evaluate all possible layouts that can be generated in this problem. In order to calculate order picking tour length for a given number of visits, one of the best known metaheuristics algorithm called Ant Colony Optimization algorithm is used. Next, Differential Evolution algorithm is used to explore the best values of the design variables to minimize average order-picking tour length for a given number of orders. Last, it was shown that two-block layouts are superior to the best-found designs with two angled and intersected cross aisles under the considered design assumptions in this study.

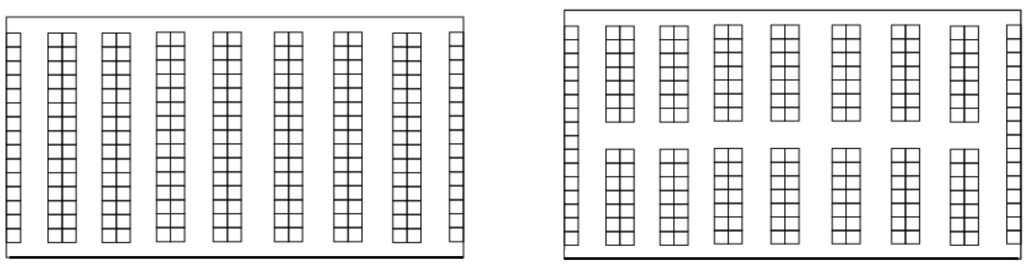
Keywords
Warehouse design, Order-picking, Aisle design, Randomized storage,

1. Introduction

Increasing online sales have been reshaping supply chain network of retail stores and operations in their warehouses. Customers may request one pencil or one AC from online stores. Considering the variety of products, their sizes and units that are requested by online customers, order-picking operation in warehouses has been increasingly playing an important role to provide competitive advantage to companies because customers expect to receive their orders quick and accurate. Thus, warehouse managers highlight that order-picking accuracy and order-cycle time are two of the important key performance metrics for managing warehouses efficiently (Warehousing Educational and Resource Council, 2018). For these metrics, order-picking operation is regarded as the most critical warehouse operation because it causes 60% of the total cycle time and 50% of the total operational cost (Frazelle and Frazelle, 2002; De Koster, 2007; Bartholdi and Hackman, 2011) when it is compared to other major warehouse operations such as receiving, put-away and shipping.

A typical order-picking operation requires order pickers to visit many locations to pick items in small quantities to fulfill customer orders. While managers perform several policies such as batching and product allocation to increase the order-picking operation's efficiency in terms of travel time, its efficiency is also affected by the layout of the storage area, which unfortunately has been undervalued for a long time by researchers because it is known that when a warehouse is built it is very difficult and expensive to change it. Because of the importance of a warehouse layout, this study aims to search for whether a new layout is generated to improve order picking operation's efficiency in warehouses.

Designing of a warehouse layout includes a series of decisions related to the number and the locations of pick-up and deposit (P&D) points, the width to depth ratios of a rectangular shape area, the number and the orientation of cross aisles, the orientation and the width of picking aisles etc. Many of the previous studies supposed that picking aisles and cross aisles are either horizontally or vertically arranged in warehouse layouts. These layouts, called traditional hereafter, are demonstrated in Figure 1. Although traditional layouts are very common in practice and academic studies, there are some studies that proposed non-traditional layouts to reduce travel distance in warehouses. The first idea of angled aisles in warehouses is conceptually suggested by Berry (1968) and White (1972). Gue and Meller (2009) improved that idea and presented a new warehouse design. With their two-innovative designs, the authors showed that the expected single command distance could be decreased about 10-20% compared to equivalent traditional designs. Using the concept of angled aisles, Öztürkoğlu et al. (2012) also proposed non-traditional warehouse layouts that present 13-20% savings in the expected-single command distance over equivalent traditional layouts. As both Gue and Meller (2009) and Öztürkoğlu et al. (2012)'s layouts are developed for centrally-located single P&D point, Öztürkoğlu et al. (2014) developed non-traditional layouts for different sets of multiple P&D point configurations using their constructive aisle model for minimizing expected single command distance. Different from these studies, Dukic and Opetuk (2008), Pohl et al. (2009) and Çelik and Süral (2014) investigated the efficiency of one of the proposed layouts, called Fishbone, by Gue and Meller (2009) for different travel commands. Pohl et al. (2009) showed that the Fishbone design presents around 10% reduction on dual-command travel compared to traditional layouts. Although Dukic and Opetuk (2008) presented that Fishbone designs cause longer travel distances for order-picking operation than Design B, Çelik and Süral (2014) showed that the average order-picking tour length could be reduced about 5-10% for small pick list sizes. The used algorithm for calculating order-picking tour length causes the reason of the difference between those studies.



a) Design A b) Design B

Figure 1. Traditional warehouse Layout examples

Because the non-traditional layout studies solely focused on Fishbone design for order-picking operation, the next section introduces a new warehouse design problem and its model. Section 3 presents our implementation of Ant Colony Optimization (ACO) algorithm for calculating order-picking tour length in a given layout. In section 4, we discussed our implementation of Differential Evolution algorithm to search for the best values of the design variables that minimize the average order-picking tour length. Section 5 presents the resulting new designs and their comparisons with traditional layouts. The last section presents concluding remarks and discussion of possible future directions.

2. The New Warehouse Layout Problem

Although previous non-traditional aisle studies considered angled aisles, their designs are restricted with non-intersecting cross aisles. In this study, we relax that assumption and introduce a new design problem. This design problem has several assumptions.

- The storage area in the warehouse has a rectangular shape. Each side of the warehouse is also a cross aisle that aims to facilitate travel between locations. They are called "periphery cross aisles".
- There are two inserted linear angled cross aisles in the storage area that intersect in the center of the warehouse. In particular, the presence of the center point is expected to facilitate travel between storage

locations. Although, they are also allowed to be originated from any side of the warehouse they are not allowed to overlap.

- Two intersecting cross aisles divide the warehouse into four picking regions. The picking aisles in each region assumed to be parallel to each other.
- There is only one P&D point in the warehouse and it is located in the middle of the front cross aisle.

Hence, we call our problem "X-shape aisle design" problem. Figure 2 shows the simple representation of X-shape non-traditional warehouse. $S_1$and S_2 refer the starting points of the first and second cross aisles, respectively. E_1 and E_2 are referred as ending points of cross aisles, respectively. Point O is the centroid of the storage area. There are four picking regions in a warehouse that is divided by the intersecting two angled cross aisles. First picking region is assumed to be the area between S_1 and S_2. Second region is specified as the area between S_2 and E_1 and the other regions are indexed in clockwise direction. Hence, angles of picking aisles in these respected picking regions defined by index $i \in \{1,2,3,4\}$ and are indicated by α_i ; $0 \leq \alpha_i \leq \pi$. Because point O is supposed to be fixed and the angled cross aisles are supposed to be linear, any X-shape warehouse layout can be represented by a vector of variables without the need of end points of the cross aisles: $\{S_{1x}, S_{1y}, S_{2x}, S_{2y}, O_x, O_y, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, where (S_{1x}, S_{1y}), (S_{2x}, S_{2y}) and (O_x, O_y) are the (x,y) coordinates of S_1, S_2 and O, respectively.

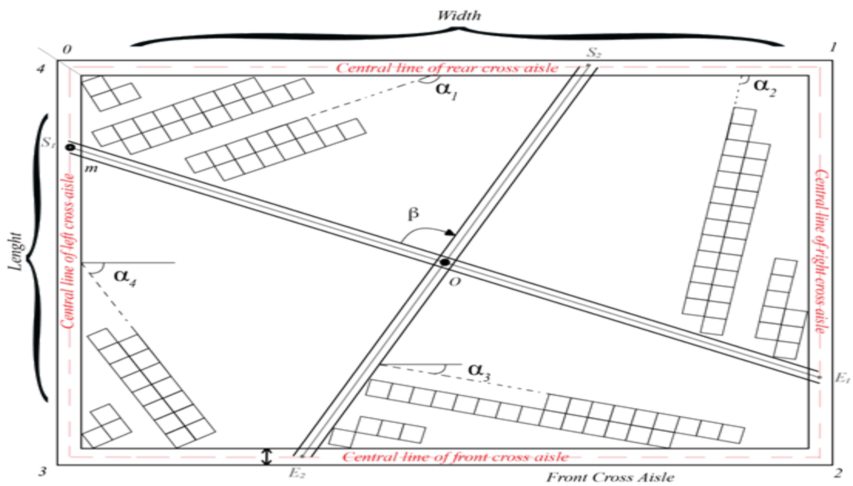


Figure 2. The demonstration of X-shape warehouse design problem

The idea of representing a general warehouse design with a vector was seen first in Öztürkoğlu et al. (2014). The authors developed a constructive aisle model to generate any layout represented by their encoding. To simplify their encoding, they used a closed loop continuous coordinate system. We also adopted their approach to simplify our encoding. In this coordinate system, the upper left, upper right, lower right and lower left corners of the rectangular storage area take the values of 0, 1, 2, 3 respectively (see Figure 2). In order to form a closed loop, the upper left corner also takes the value of 4. Hence, the variable $m \in [0,4]$ is used to indicate the initial point of first angled cross aisle, which also replaces S_1. Moreover, le β t be the clockwise angle ($0 \leq \beta \leq \pi$) between the first and the second angled cross aisles. When m and β is known, the initial point of the second angled cross aisle can easily be obtained, as well as its end. Last, the encoding is reduced to $\{m, \beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. We also studied on the symmetric layouts based on the orientation of the angled cross aisles. Thus, we can reduce the search space of $m \in [0,2]$.

In the constructive aisle model of Öztürkoğlu et al. (2014), after a layout is generated for a given encoding, its network is developed so that travel distances between storage locations and P&D points could be calculated. Although our problem differs from Öztürkoğlu et al. (2014)'s problem due to intersecting cross aisles in the center, we adopted their approach because the concept is similar. Hence, the layout is simply generated using their procedure for a given encoding.

After a layout is generated, we develop its network. A typical warehouse network consists of a set of special nodes called access nodes, pick-up and deposit nodes, travel nodes and cross nodes (see Figure 3 for demonstration). Access nodes are the points that are located on the central line of a picking aisle. As seen in the figure, even though some access nodes, of which one serves the pallet location on the left and the other serves to the pallet location on the opposite side, are located on the same coordinate, however they are uniquely defined for the purpose of accurate network representation. In order to allow travel between these nodes, they are connected by an edge with a distance of zero. The centers of the pallet locations are used to calculate the coordinates of these access nodes. The pick-up and deposit nodes represent the existing P&D points from and to where materials go through. Travel nodes are intersecting points of central lines of picking and cross aisles that are assumed to be used to access to the picking aisles. Last, cross nodes are defined as the intersects of central lines of cross aisles that are used to change aisles for ease of travel to the required locations. For the sake of clarification, only cross nodes that are made from intersecting side cross aisles shown on the figure. Last, appropriate nodes are connected by edges with a weight of distance between the connected nodes. We assume that edges are undirected that allow two-way travel with the same distance. Once the network of a layout is generated, it is easy to calculate travel distances between locations.

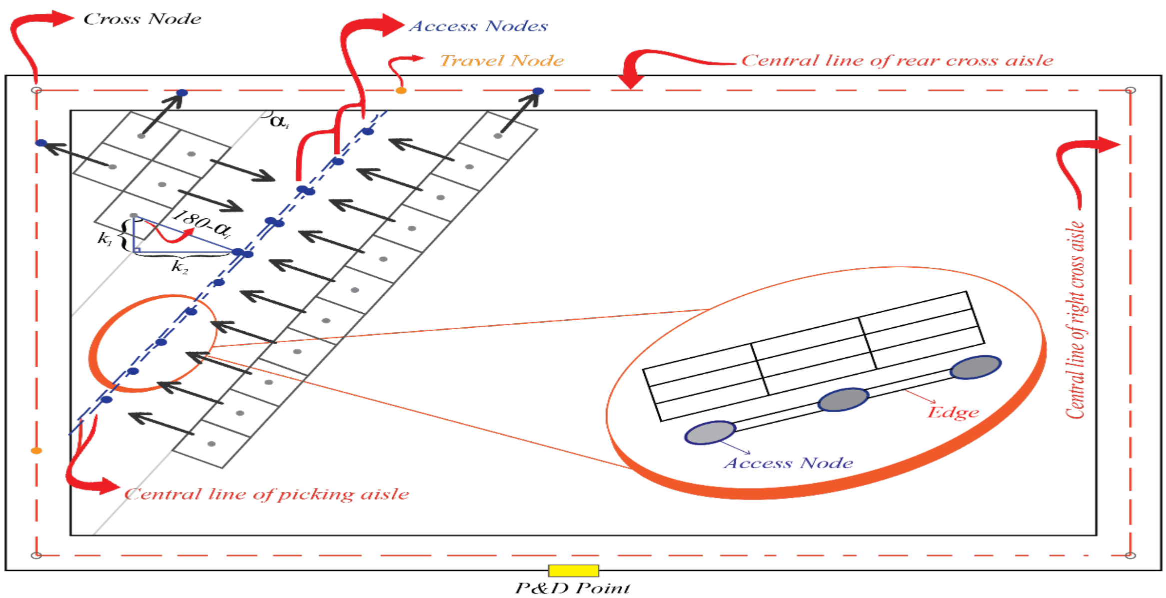


Figure 3. Travel nodes, access nodes and cross nodes representation

3. Order-Picking Tour Length

An order picker route is the path that is constructed by the sequence of locations that need to be visited by the picker according to a given pick list. A pick list consists of required items and their locations to be visited by a picker. A picker visits these locations in a tour starting and ends at a P&D point. Hence, the length of a tour is the total distance required to visit these locations starting from P&D point and ending at the P&D point. Last, this problem resembles to well-known TSP problem and the aim is to find the shortest route length for order picking operation.

In order to solve order-picking routing problem in warehouses, exact algorithms and heuristics were developed. In traditional layouts, Ratliff and Rosenthal (1983), Roodbergen and De Koster (2001a), Gelders and Heeremans (1994), Roodbergen and de Koster (2001b) and Öztürkoğlu and Hoşer (2018) developed exact approaches to solve optimal order-picking tour lengths for given orders. In addition to these exact algorithms, special heuristics, such as s-shape, aisle-by-aisle, largest-gap, mid-point, and return, have been proposed for generating reasonable routes in traditional warehouse designs (Kunder and Gudehus, 1975; Hall, 1993; Petersen, 1997; Roodbergen and de Koster, 2001b). Because many of these algorithms either work well or are specifically designed for traditional layouts, we prefer to use a metaheuristic algorithm.

For our model's route length calculation, we preferred the Ant Colony Optimization (ACO) algorithm, which was firstly designed by Dorigo and Gambardella (1997) for the discrete TSP problem. This metaheuristic algorithm is inspired by acting ants in the nature. Because ants secrete a hormone, which is called pheromone, they can find their direction. The ants leave the pheromones on the roads they are passing through and choose from alternative routes as they return to the point of food or nest. These pheromones actually a sign for other ants. The choice made depends on the distance of the roads and the amount of pheromone on the route. In the alternative routes if the pheromone amount is equal, more ants began to prefer shorter routes. Then, the pheromone amount on the shortest route increases that results in favouring the shortest route.

The algorithm starts with c number of ants that are randomly assigned to a point. At each step, an ant determines its next point probabilistically, depending on the distance and the amount of pheromone. In tour t, ant k can travel between i to j points in a probability of $p_{ij}^k = \frac{[\tau_{ij}]^{PR}[\eta_{ij}]^{VR}}{\sum_{I \in N_i^k}[\tau_{ij}]^{PR}[\eta_{ij}]^{VR}},\ if\ i\ \in N_i^k$. In this function τ_{ij} represents the pheromone amount between points i and j. η_{ij} is visibility intuitive value, which is the reverse of distance between i and j points ($1/d_{ij}$). PR and VR are variables that determine the pheromone and visibility intuitive's relative effect according to decision function. For example; $PR = 0$ means that pheromone amount is ignored, just visibility intuitive is important. $VR = 0$ means that selection is made only based on the amount of pheromone. N_i^k is a set of unvisited point of ant k when it is located at point i. The best route which is made by ants visiting for all points until that time is kept. The ants continue to form routes depending on the amount of pheromone until stopping criterion is reached.

Once we obtained the best route in every iteration, we apply 2-opt local search algorithm to improve the solution. This algorithm is for shifting the two consecutive links to search for better solutions. We prefer this algorithm because it is easy to implement and efficient way to improve a solution. In our implementation, if a solution is improved apter applying the 2-opt local search algorithm, we updated the best tour. In case there is no improvement, the ACO algorithm continues with the current best solution. For the distances between picking locations and the P&D points, we use Dijkstra's shortest path algorithm for a given pick list.

4. Design Optimization: Differential Evolution (DE) Algorithm

In this stage, the route will be calculated according to the pick list given in a warehouse in which the layout is enhanced and the network is created. The average route length is calculated for an order consisting of a certain number of collection points. This can be thought of as the fitness value of the warehouse. As we mentioned in the previous section, the aim is to minimize this fitness. The main problem in achieve this aim is the best values of the variables in the encoding. These variables are continuous variables. In order to optimize variables, we prefer one of the metaheuristic algorithms that showed efficacy in continuous variable optimization; Differential Evolution Algorithmb.

Differential Evolution Algorithm (DE) is proposed by Storn and Pierce (1997). The simple DE algorithm is an evolutionary algorithm. Like genetic algorithms (GA) and evolutionary strategies, it can be used to improve the results. This is one of the metaheuristic algorithms which is a population-based and stochastic search tool. Due to this tool can be applicable to both discrete and continuous optimisation problems, it is frequently used in recent researches. Its simple structure, good results in complex problems, ease of implementation and robustness are the advantages of a basic DE algorithm (Jitkongchuen and Thammano, 2014). For continuous problems, according to Kitayama et al., (2011) the results of comparison between DE and Particle Swarm Optimisation (PSO) showed that DE algorithm is more efficient than PSO algorithm.

The algorithm must be followed by four main steps and these steps are shown in Figure 4. With the first step as initialization, the algorithm starts to optimize according to variables. We denoted the population size as P for implementation and G is used as generation number. So, the vector with D-dimension can be represented as $X_{i,G} = [x_{i,G}^1, x_{i,G}^2, \ldots, x_{i,G}^D]$, $i = 1,2, \ldots, P$. To cover all the search space, parameters' upper and lower bounds should be defined. After definition of boundaries, first parameter of value selection is made randomly from boundaries. The initialization phase completes after each P parameter in the vector is defined.

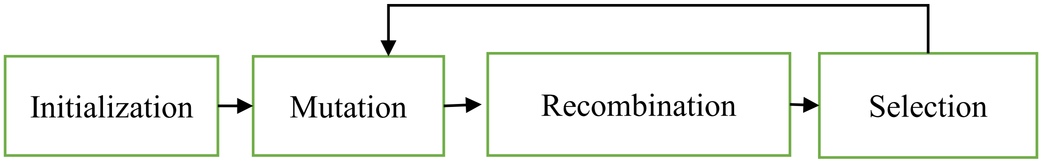


Figure 4. Main steps of DE algorithm

The second step as mutation, is used for expanding the search area. For given $X_{i,G}$ parameter, under $i \neq r_1, r_2, r_3$ situation, three different parameters $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ are selected randomly. With $V_{i,G+1} = X_{r1,G} + F\left(X_{r2G} - X_{r3,G}\right)$ formulation, a mutation vector is created. In this formulation, F is a mutation factor and $V_{i,G} = \left\{v_{i,G}^{1}, v_{i,G}^{2}, \ldots, v_{i,G}^{D}\right\}$ vector represents the donor vector.

In order to optimise our design problem in the mutation step, two different strategies were integrated in DE. These strategies are described in the following formulas. During the mutation operation, each strategy is applied with equally likely. The decision of which parameter is going to be use in mutation is chosen randomly in the range $[1, P]$. $X_{best,G}$ is the best vector which has the best fitness at generation G.

- DE/ Best/1: $V_{i,G} = X_{best,G} + F\left(X_{r_1^i,G} - X_{r_2^i,G}\right)$ (Storn, 1996)
- DE/Rand/2: $V_{i,G} = X_{r_1^i,G} + F\left(X_{r_2^i,G} - X_{r_3^i,G}\right) + F\left(X_{r_4^i,G} - X_{r_5^i,G}\right)$ (Qin et al., 1997)

The recombination step involves successful solutions of the previous generation. Donor vector elements are included in the trial vector ($U_{i,G} = \left\{u_{i,G}^{1}, u_{i,G}^{2}, \ldots, u_{i,G}^{D}\right\}$) with crossover rate (CR) probability. DE uses a uniform recombination as in defined below (Storn and Price, 1997).

$u_{i,G}^{j} = \begin{cases} V_{j,i,G+1} & if\ \left(rand_{j,i} \sim U[0,1]\right) \leq CR\ or\ (j = j_{rand}) \\ x_{j,i,G+1} & otherwise \end{cases}$, $j = 1,2, \ldots, D$, j_{rand} is a random integer from [1, 2,.., D].

In the selection process, which is the last step of DE, compares the target vector $X_{i,G}$ and trial vector $u_{i,G+1}^{j}$. It is ensured that the better value of the function is transferred to the next generation. With the equation $X_{i,G+1} = \begin{cases} U_{i,G}, & if\ f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & otherwise \end{cases}$, $i = 1,2, \ldots, P$, if the trial vector has lower function value then this trial vector is selected as a next generation. If it is the opposite, then the target vector moved to next generation. The DE algorithm performs continuously with mutation, recombination and selection steps until termination criterion is realized.

As we explained in section 2, there are six variables of our problem. m is bounded with [0,2], β is defined as in the range of $[0, \pi]$ and α_i's can change in the range of $[0, \pi]$ for every $i \in \{1,2,3,4\}$. While searching the best value in the DE algorithm, these variables may go out of their bounds during mutation. In order to keep them in their boundaries, we use the Periodic Approach which is proposed by Padhye et al. (2015) as a constraint-handling strategy. This strategy bounds the constraints with a periodic repetition ($p = x^{(U)} - x^{(L)}$) of the objective function. With equation $y = \begin{cases} x^{(U)} - \left(x^{(L)} - x^{C}\right) \%\, p\,, & if\ x^{C} < x^{(L)}, \\ x^{(L)} + \left(x^{C} - x^{(U)}\right) \%\, p\,, & if\ x^{C} > x^{(U)} \end{cases}$, (where % is used for mode operation) a breached variable is turned into boundaries of $[x^{(L)}, x^{(U)}]$; and become a new variable noted as y. For our problem; $x = \{m, \beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$.

There are many parameters that need to be set in DE. In this process, Mallipeddi et al.'s (2011) study on parameter settings and mutation strategies was used. We take P as 50. According to Mallipeddi et al. (2011), CR should be taken in the range [0.1-0.9] in order to balance efficiency and speed. Thus, we take CR as 0.5. Similarly, we take F as 0.4 because it is mentioned that F should be taken in the range [0.4-0.9].

We use iteration number (1000 iterations) to terminate our algorithm with a consdition of no improvement in the last 100 iterations. Hence, if there is a chance to improve the solution, we run the algorithm a bit more. According to our stopping criteria; first 1000 iteration are completed without any condition. At every iteration after 1000 iterations, the best solution is compared with the previous best solution. If there is no improvement, then the algorithm stops at iteration 1100. However there is an improvement, the algorithm runs extra 100 iterations unless there is no further improvement. Last, the final best solution is taken as the best-solution found so far.

5. Numerical Study and Results

This section describes the search for the best X-shape warehouse layout where average tour length is minimized. We consider a warehouse layout twice as wide as its deep. We assume single deep racking system. We assume that all products stored under randomized storage policy because of its simplicity, its more efficient use of storage space and popularity in the industry (Petersen, 1999). Therefore, we assume uniform picking. We also locate the single P&D point in the middle of the front cross aisle because Roodbergen and Vis (2006) showed that this is the optimal location for a single P&D point that minimizes the order-picking tour under randomized storage.

In order to consider the effect of number of picks on average tour length, we use seven pick list sizes (3, 5, 10, 20, 30, 40 and 50) (Çelik and Süral, 2014). For example; if the pick list size is 30, order picker travels 30 different locations in a warehouse starting from the P&D point and ends the travel at the P&D point. So, if there are m number of storage locations on warehouse and the picking list size is n, $C(m,n)$ number of different order lists could be created. Because of considering every possible order and its tour length will be incredibly time consuming, we statistically determine an appropriate number of orders. First, we generate 1000 orders for both small (3) and large (50) pick lists to statistically analyze the appropriate sample size. We calculate our sample size using the formula $N = \frac{s^2 \cdot Z_{\alpha/2}^2}{(\bar{x} \cdot 0.01)^2}$ where s is the standard deviation, $\bar{x}$ is the sample mean. As a result we are 95% confident with a %1 of error that 1500 orders are appropriate for smaller pick lists such as 3,5 and 10, and 250 orders for larger pick lists such as 20, 30, 40 and 50.

The arithmetic average of all orders' tour length is taken as the fitness value or cost of the designs. As explained in previous section, we use ACO algorithm to calculate an order picking tour length and we use DE algorithm to search for the best value of design variables to minimize average tour length. In order to compare the best-found designs, we take Design B as a base. The considered Design B in this study has 11 vertical picking aisles, one middle cross aisle in the center, its width and heights are 580 and 290 pallet units. Hence, there are 512 storage locations available at the lowest level of the racks. We also take the same width and depth for X-shape design.

We run our experiments on a computer running on 4 GB RAM and a 1.70 GHz Intel ® CORE i5 processor. We perform three replications. We then select the one with the best fitness value. The average computational time for these replications are shown in Table 1. Table 2 shows the results of the best X-shape designs found so far for each pick list. This table also shows the average tour length of Design B for the same order set. Figure 5 presents the comparison of the best-found X-shape designs and Design B when their warehouse sizes are equal. As seen in the figure, the best-found X-shape designs provide about 5% less average tour length with a cost of 30% loss in capacity for equivalent sizes of 2:1 (width-to-depth ratio) warehouses. The reasons of the loss in space are simply the inserted additional cross aisles and angled aisles.

Table 1. Average Running Times

Pick list size	3	5	10	20	30	40	50
CPU time (hr)	4.6	11.2	24.8	27.9	61.7	90.2	125.5

Table 2. The solutions for the best-found X-shape designs and the fitness of Design B

Shape ratio 2:1, Width 580 PU, Length 290 PU									
Pick List #	Capacity	m	β	α_1	α_2	α_3	α_4	X-shape Fitness	Design B Fitness
3	383	1.56	86.9	110.8	65.3	70.2	128.3	1,012.7	1,082.4
5	357	1.23	28.3	109.5	85.0	78.5	98.9	1,306.4	1,358.2
10	369	1.80	152.5	85.4	106.5	94.4	79.2	1,798.7	1,839.3
20	337	1.08	36.9	78.5	89.0	76.6	81.1	2,405.5	2,521.4
30	345	1.08	33.5	79.9	87.5	84.8	86.5	2,831.9	2,978.9
40	346	1.29	35.0	84.4	94.3	100.6	98.2	3,168.2	3,313.2
50	374	1.86	136.1	88.9	95.8	81.8	93.3	3,475.7	3,558.0

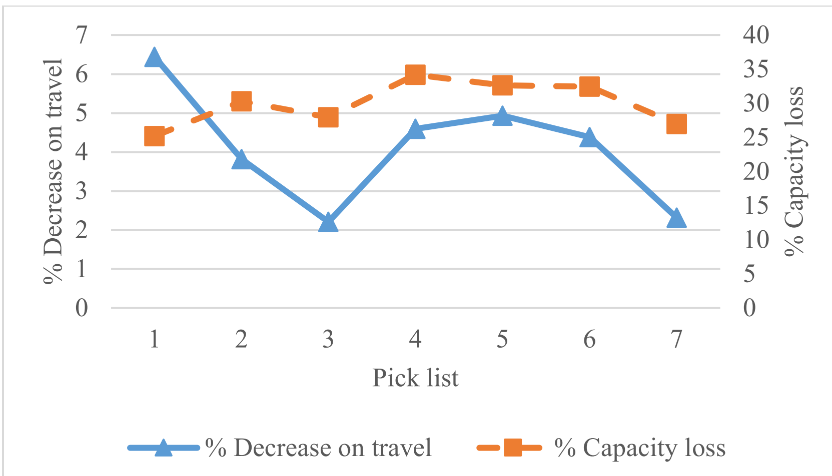


Figure 5. Comparison of the best-found X-shape design and Design B when the warehouse sizes are equivalent

In order to provide an accurate comparison, the size of X-shape designs are expanded to provide same amount of storage locations with an equivalent Design B. While expanding the warehouse sizes, the obtained values in the best solution including α_is and m are kept fixed, as well as shape ratio. Table 3 summarizes the characteristics of these expanded best-found X-shape designs. When these expanded designs are compared with the equivalent Design B, we see that shape designs do not provide any savings on travel (see Figure 6). Additionally, they require almost 25% more storage area than the equivalent Design B to occupy same number of storage locations. Last, the best-found X-shape designs can be seen in Figure 7.

Table 3. Features of the expanded best-found X-shape designs

Shape Ratio	Width	Length	Area	Pick List Size	Capacity	Average Tour Length
2:1	620	310	192.2	3	500	1,104.3
	640	330	211.2	5	520	1,464.3
	640	320	204.8	10	509	1,992.7
	650	330	214.5	20	514	2,808.3
	640	330	211.2	30	515	3,356.1
	640	330	211.2	40	507	3,747.1
	640	330	211.2	50	505	4,144.8

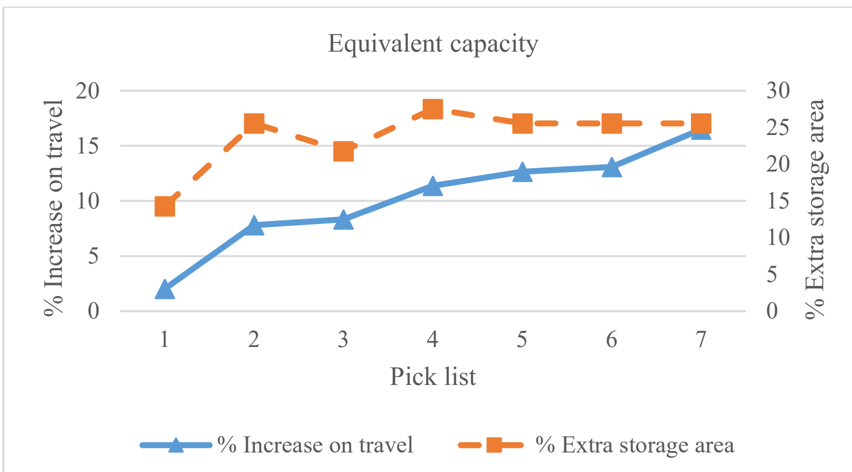
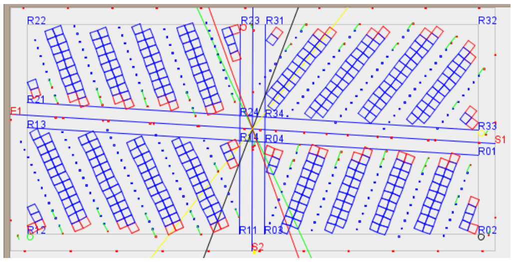


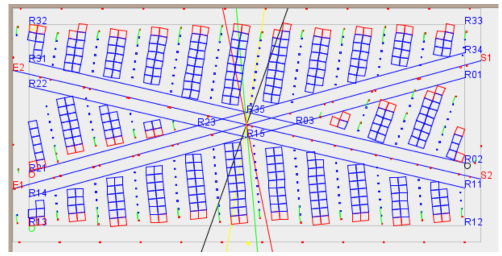
Figure 6. Comparison of Expended X-shape and Design B in Same Capacity

6. Conclusion

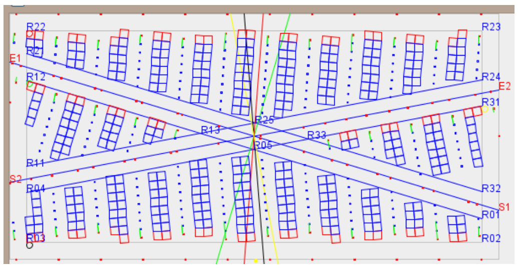
In this study, we study on a completely new non-traditional warehouse design problem. In this problem, we inserted two angled cross aisles in a rectangular shape warehouse. We assume that these angled cross aisles intersect in the middle of storage area. Additionally, the picking aisles are also assumed to take any angle. We then search for the best designs where there is only one P&D point that allows material flows in and out, which is located on the middle of front cross aisle. We also assume that picking from any location has equal chance under randomized storage policy. After searching the best designs in warehouses with 2:1 width to shape ratio for different pick list sizes, we see that any of the best-found X-shape designs could reduce average order-picking tour length over the equivalent Design B. One of the reasons of this result is that the best-found designs require approximately 25% larger storage area than Design B due to angled and inserted aisles. Because of the limitations of this study with uniform picking, 2:1 shape ratio and single P&D point, further investigation can be performed to look for the existence of better designs in this new problem.



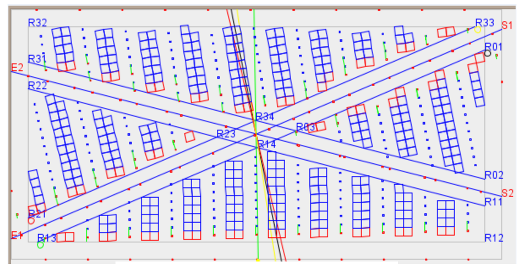
For 3 pick



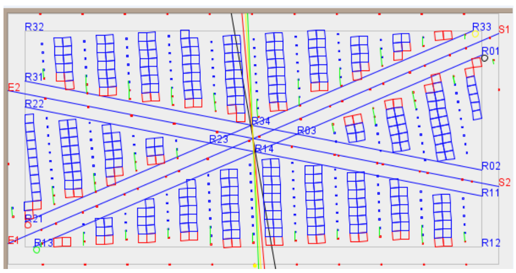
For 5 pick



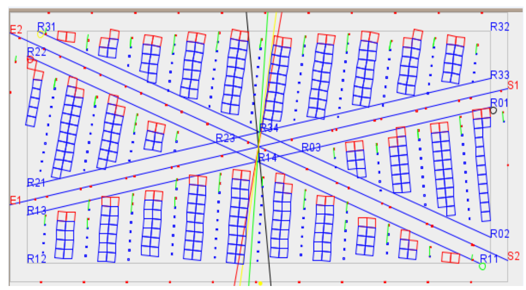
For 10 pick



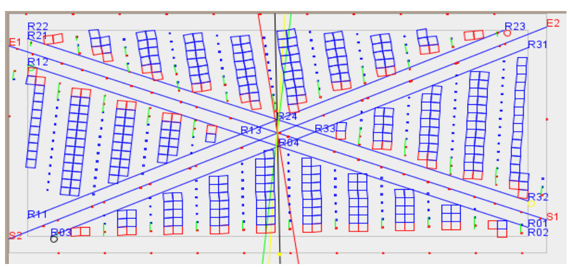
For 20 pick



For 30 pick



For 40 pick



For 50 pick

Figure 7. The best-found X-shape design representations when the shape-ratio is 2:1

Acknowledgements

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant No: 214M220. This work is a part of the master thesis studied by Aysegul Mağara.

References

Bartholdi, J. J., and Hackman, S. T., Warehouse and Distribution Science: Release 0.95, *The Supply Chain and Logistics Institute*, School of Industrial and Systems Engineering. Atlanta, 2011.

Berry, J. R., Elements of warehouse layout. *The International Journal of Production Research,* vol. 7, no. 2, pp. 105-121, 1968.

Çelik, M., and Süral, H., Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE transactions*, vol. 46, no. 3, pp. 283-300, 2014.

De Koster, R., Le-Duc, T., and Roodbergen, K. J., Design and control of warehouse order picking: *A literature review. European journal of operational research*, vol. 182, no. 2, pp. 481-501, 2007.

Dijkstra, E. W., A note on two problems in connexion with graphs. *Numerische mathematik*, vol. 1, no. 1, pp. 269-271, 1959.

Dorigo, M., and Gambardella, L. M., Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53-66, 1997.

Dukic, G., and Opetuk, T., Analysis of order-picking in warehouses with fishbone layout. *Proceedings of ICIL*, vol. 8, 2008.

Frazelle, E., and Frazelle, E., *World-class warehousing and material handling* (vol. 1). New York: McGraw-Hill, 2002.

Gelders, L., and Heeremans, D., Het travelng salesman probleem toegepast op order picking. *Tijdschrift voor economie en management*, vol. 39, no. 4, 1994.

Gue, K. R., and Meller, R. D., Aisle configurations for unit-load warehouses. *IIE Transactions*, vol. 41, no. 3, pp. 171-182, 2009.

Hall, R. W., Distance approximations for routing manual pickers in a warehouse. *IIE Transactions*, vol. 25, no. 4, pp. 76-87, 1993.

Jitkongchuen, D., and Thammano, A., A self-adaptive differential evolution algorithm for continuous optimization problems. *Artificial Life and Robotics*, vol. 19, no. 2, pp. 201-208, 2014.

Kitayama, S., Arakawa, M., and Yamazaki, K., Differential evolution as the global optimization technique and its application to structural optimization. *Applied Soft Computing*, vol. 11, no. 4, pp. 3792-3803, 2011.

Kunder, R., and Gudehus, T., Mittlere wegzeiten beim eindimensionalen kommissionieren. *Zeitschrift für Operations Research*, vol. 19, no. 2, pp. B53-B72, 1975.

Mallipeddi, R., Suganthan, P. N., Pan, Q. K., and Tasgetiren, M. F., Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied soft computing*, vol. 11, no. 2, pp. 1679-1696, 2011.

Öztürkoğlu, Ö., Gue, K. R., and Meller, R. D., Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, vol. 44, no. 6, pp. 459-475, 2012.

Öztürkoğlu, Ö., Gue, K. R., and Meller, R. D., A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points. *European Journal of Operational Research*, vol. 236, no. 1, 382-394, 2014.

Öztürkoğlu, Ö., and Hoşer, D., A new warehouse design problem and a proposed polynomial-time optimal order picking algorithm. *Journal of the Faculty of Engineering and Architecture of Gazi University*, https://doi.or./10.17341/gazimmfd.416453, in press, 2018.

Padhye, N., Mittal, P., and Deb, K., Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Computational Optimization and Applications*, vol. 62, no. 3, pp. 851-890, 2015.

Petersen, C. G., An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, vol. 17, no. 11, pp. 1098-1111, 1997.

Petersen, C. G., The impact of routing and storage policies on warehouse efficiency. *International Journal of Operations & Production Management*, vol. 19, no. 10, pp. 1053-1064, 1999.

Pohl, L. M., Meller, R. D., and Gue, K. R., Optimizing fishbone aisles for dual-command operations in a warehouse. *Naval Research Logistics (NRL),* vol. 56, no. 5, pp. 389-403, 2009.

Qin, A. K., Huang, V. L., and Suganthan, P. N., Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation,* vol. 13, no. 2, 398-417, 2009.

Ratliff, H. D., and Rosenthal, A. S., Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, vol. 31, no. 3, pp. 507-521, 1983.

Roodbergen, K., and de Koster, R., Routing Order Pickers in a Warehouse with a Middle Aisle. *European Journal of Operational Research,* vol. 133, pp. 32-43, 2001a.

Roodbergen, K. J., and Koster, R., Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research,* vol. 39, no. 9, pp. 1865-1883, 2001b.

Roodbergen, K. J., and Vis, I. F., A model for warehouse layout. *IIE transactions*, vol. 38, no. 10, pp. 799-811, 2006.

Storn, R. (). On the usage of differential evolution for function optimization. In Fuzzy *Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American* (pp. 519-523). IEEE, 1996.
Storn, R., and Price, K., Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, vol. 11, no. 4, pp. 341-359, 1997.
Warehousing Education and Research Council, 2018 DC Measures Study, USA, 2018.
White, J. A., Optimum design of warehouses having radial aisles1. *AIIE Transactions*, vol. 4, no. 4, pp. 333-336, 1972.

Ömer Öztürkoğlu is an Assistant Professor in Business Administration at Yasar University, Izmir, Turkey. After completing his undergraduate studies at the Department of Industrial Engineering in Dokuz Eylul University, Turkey, he completed his Master studies at Erciyes University from the Industrial Engineering Department in Turkey. Next, he completed his master and doctoral studies in Industrial and Systems Engineering at Auburn University, Alabama, USA. He teaches related courses to Logistics Facilities, Warehousing, and Production and Operations Analysis. In general, his research interests are warehousing, production and operations systems analysis and design.

Ayşegül Mağara is currently managing a project in Schneider Electric, İzmir, Turkey. She has gained experience in main departments such as production and supply chain for nearly 3 years business life. Miss. Magara holds both Bachelor and Masters of Science degree in Industrial Engineering from Yaşar University. She took part in a TUBITAK project while in a Master's degree.