YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

# A NEW WAREHOUSE LAYOUT MODEL WITH INTERSECTING NON-TRADITIONAL CROSS AISLES

AYŞEGÜL MAĞARA

THESIS ADVISOR: ASST.PROF.DR. ÖMER ÖZTÜRKOĞLU

INDUSTRIAL ENGINEERING

PRESENTATION DATE: 22.05.2018

BORNOVA / İZMİR
MAY 2018

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

| **Jury Members:** | **Signature:** |
|---|---|
| Asst. Prof. Dr. Ömer ÖZTÜRKOĞLU<br>Yaşar University | ........................ |
| Asst. Prof. Dr. Adalet ÖNER<br>Yaşar University | ........................ |
| Asst. Prof. Dr. Ceyhun ARAZ<br>Celal Bayar University | ........................ |

---------------------------------------------------------------

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

# ABSTRACT

A NEW WAREHOUSE LAYOUT MODEL WITH INTERSECTING NON-TRADITIONAL CROSS AISLES



Mağara, Ayşegül

Msc, Industrial Engineering

Advisor: Asst. Prof. Dr. Ömer ÖZTÜRKOĞLU



May 2018



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, eventually leading to the emergence of non-traditional layouts. In this thesis, 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, therefore this design idea is called as X-shape warehouse layout. A new constructive aisle model is developed so as to evaluate all possible layout options that can be generated by the new idea. In order to calculate order picking tour length, 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 the best-found X-shaped designs unfortunately do not provide any savings on tour length over the equivalent two-block traditional designs in many cases. Only best-found designs with 3:1 shape ratios presented 1.5% reduction on average tour length for small picklist sizes in comparison to the equivalent two-block layouts.



**Key Words:** Non-traditional warehouse design; Aisle design; Randomized storage; Constructive algorithm; Reel number optimisation

# ÖZ

KESİŞEN ANA KORİDORLU GELENEKSEL OLMAYAN YENİ BIR DEPO TASARIMI

Mağara, Ayşegül
Yüksek Lisans Tezi, Endüstri Mühendisliği
Danışman: Öğr. Üyesi Dr. Ömer ÖZTÜRKOĞLU

Mayıs 2018

Yakın zamanda yapılan depo tasarımları, sipariş toplama işlemlerinin tur mesafesinin geleneksel depo tasarımlarındaki koridorların açısını değiştirerek azaltılabilmesine ve geleneksel olmayan (yenilikçi) tasarımların ortaya çıkmasına yol açmıştır. Bu tezde, sipariş toplama operasyonu için geleneksel iki bloklu depo tasarımlarından daha iyi bir dizayn arayışında, yeni bir tasarım fikri önerilmiştir; depolama alanının ortasında iki ana koridorun kesişmesine izin verilmiştir ve bu tasarım X şeklindeki depo tasarımı olarak adlandırılmıştır. Bu yeni fikir tarafından üretilebilecek tüm olası yerleşim seçeneklerini değerlendirmek için yeni bir koridor modeli geliştirilmiştir. Sipariş toplama için yapılan tur uzunluğunu hesaplamak için çok bilinen bir meta-sezgisel yöntem olarak Karınca Kolonisi Optimizasyon Algoritması kullanılmıştır. Sonrasında Diferansiyel Evrim Algoritması, verilen sipariş listesi uzunluğuna göre ortalama sipariş toplama tur uzunluğunu en aza indirmek için tasarlanan modelin değişkenlerinin en iyi değerlerini araştırmak için kullanılmıştır. Son olarak, maalesef ki çoğu durumda X şekilli tasarımların eşdeğer iki bloklu geleneksel tasarımlar üzerinde tur uzunluğunda herhangi bir tasarruf sağlamadığı gösterilmiştir. Elde edilen 3:1 şekil oranına sahip X şeklindeki en iyi depo tasarımlarında, eşdeğer iki bloklu tasarıma kıyasla küçük sayıdaki sipariş toplama işlemi için ortalama tur uzunluğunda %1.5 azalma sağlanmıştır.

**Anahtar Kelimeler:** Yenilikçi depo tasarımı; Koridor tasarımı; Rassal depolama politikası; Sipariş toplama rota uzunluğu; Çözüm oluşturma algoritması; Reel sayı en iyileme

## ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Ömer Öztürkoğlu for his guidance and patience during this study. The door to Asst. Prof. Dr. Öztürkoğlu office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it.

I must express my very profound gratitude and enduring love to my parents and to my boyfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Without your endless patience, unconditionally love, and encouragement, I wouldn’t be here today. Thank you.

This thesis was supported by the TUBİTAK (The Scientific and Technological Research Council of Turkey) within the scope of the Grant No. 214M220.

Ayşegül Mağara

İzmir, 2018

## TEXT OF OATH

I declare and honestly confirm that my study, titled "A New Warehouse Layout Model with Intersecting Non-Traditional Cross Aisles" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Ayşegül Mağara

Signature

..........................................

June 12, 2018

# TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

TSP Travelling Salesman Problem

ACO Ant Colony Optimisation Algorithm

DE Differential Evolution Algorithm

P&D Pick-up and Deposit

SYMBOLS:

L Length of warehouse.

H Height of warehouse.

a The width of the picking aisles.

$S_i$ Starting point of aisle $i$

$E_i$ Ending point of aisle $i$

$m$ First cross aisles starting point

$\propto_i$ Angle of picking aisle in area $i$

$\beta$ Angle between two cross aisles

$c$ Number of ant

$PR$ Pheromone rate

$VR$ Visibility rate

$ER$ Evaporation rate

P Population size

CR Crossover Rate

F Scaling Factor

# CHAPTER 1
# INTRODUCTION

E-commerce activities and online sales are increasing day by day in today's world. According to an annual survey that was made by comScore and UPS, consumers buy 51% of their needs from online stores. The percentage was 47% back in 2014 (Farber, 2016). In order to meet this increasing demand, online stores were started to demonstrate larger storage areas. One of the key factor of making consumers satisfied is delivering their products as quick as possible. Consequently, warehouses have started playing an important role in the supply chain management because the importance of customer satisfaction and response time to customer orders have been increasing. Therefore, the duration to perform warehouse operations become one of the critical performance measures for warehouse managers. The major warehouse operations are receiving, put-away, order-picking and shipping. Among these operations, the order-picking operation is regarded as the most costly and critical operation because it causes 60% of the total cycle time and 50% of the total operational cost (Frazelle, 2002; De Koster, 2007; Bartholdi and Hackman, 2011). In order to decrease the order-picking operation cost, mainly 3 approaches are defined in the literature; namely using batches, changing layout and changing storage policy. Order batching is basically grouping orders to be simultaneously picked in a single tour and the aim is finding the optimum way of grouping to minimize picking time or travel length. According to specifications of products in the warehouse like consumption frequency and size, using appropriate storage policy can also reduce travel time or picking length. The storage policy types are briefly mentioned in Chapter 1.1.2. Also there are many studies in the literature for both batching and storage policy (Henn and Wascher, 2010; Chackelson et al., 2011; Rouwenhorst et al., 2000; Hong et al., 2012). Another variation of the approach is the layout change, which has a significant effect on the order picking process. We can see in the literature the studies on traditional designs which have been widely used for many years. But also in recent years there are studies which based on the idea of changing the layout to different shapes.

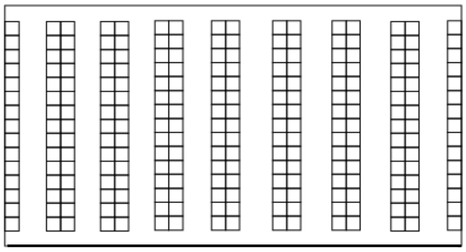
Since there are researches a few in number that has focused on order-picking operations in the non-traditional aisles, this thesis aims to fill this gap in the literature (Dukic and Opetuk, 2008; Çelik and Süral, 2013). In this thesis, we focused on the way possibility of changing the layout. There is a kind of warehouse layout that is commonly used named as traditional warehouse which is easy to implement and the efficiency of tour length is quite remarkable. Therefore, the question is if the main aisles are changed like the way never studied before, could order picking tour length be reduced? According to rule of triangle inequality, the hypotenuse is shorter than the sum of the other two sides in a triangle. With respect to this law, we inset two intersecting cross aisles in storage area, which allows to make the picker less total tour length than traditional ones under same order picking list. In this thesis, we research optimal warehouse layout with X-shape cross aisles under random storage policy with modelling it and analyze its efficiency for order-picking operation.

The reminder of this thesis is organized as follows: We review the related literature in Chapter 1. In Chapter 2, we present a new warehouse design and define the modelling of our problem. The calculation of route length in the warehouse and implementation of related algorithms is detailly explained in the Chapter 3. We finally provide results of our problem, an overall summary of this thesis and discuss possible directions for future research in Chapter 4.

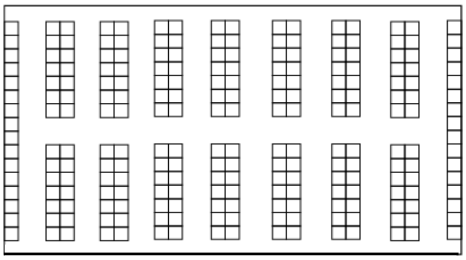
## 1.1. Literature Review

### 1.1.1. Aisle Design

Designing of warehouse layouts includes a series of decision. Such as “Where to a Pick-up and Deposit (P&D) point will be represented”, “How many blocks and aisles should be there”, “Where the main aisle is going to be represented”. Although there are few studies in the literature, changing layout design may provide significant efficiency (Gue and Meller, 2009b; Öztürkoğlu et al., 2014; Mesa, 2016). It can be argued that the layout type named as traditional warehouse that was generated in nearly half century ago having been the most common layout type. Gue and Meller (2009) referred a layout with highlighting two main design rules; first, order picking aisle should be parallel to each other (horizontal or vertical) and second if there are main aisles then they should be perpendicular to order picking aisles. There are examples of traditional layout types represented in Figure 1.1. These designs are the most common in warehouses, because they are less costly, easy to be implemented and they provide many efficiently usable places for storage.



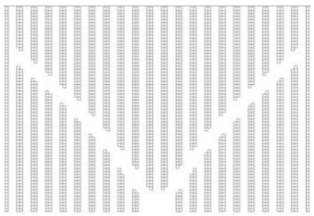
a) Design A



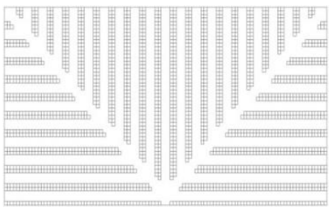
b) Design B

**Figure 1.1** Traditional Warehouse Layout Examples.

Although traditional layouts are commonly used, in recent years there are some studies that propose non-traditional layouts to reduce travel distance in warehouse. The first idea of angled aisles in warehouses is suggested by Berry (1968) and White (1972) but they didn't propose any layout. Gue and Meller (2009) improved that idea and converted it to a new warehouse design. With their two-innovative designs (shown in Figure 1.2), the authors showed that expected single command distance could be decreased about 10-20% compared to the equivalent traditional design.



a) Flying-V



b) Fishbone

**Figure 1.2** Examples of Innovative Warehouse Designs. Adapted from: "Aisle configurations for unit-load warehouses." By Gue, K. R., & Meller, R. D. (2009), IIE Transactions, 41(3), 171-182.

Pohl et al. (2009b) studied the innovative Fishbone design for dual-command operation and show that this design reduces dual-command travel by 10–15%. In Pohl et al. (2011) they investigate both Flying-V and Fishbone design within the scope of the turnover based storage policy, and they also showed that these layouts still provide reduction on travel distance over traditional layouts. Öztürkoğlu et al. (2012) proposed innovative designs for warehouses which have single command operation and random storage policy: Chevron, Leaf and Butterfly (shown in Figure 1.3). For Chevron design, it is noted that average single-command distance decreased about 13-20%, however area requirement increased in average 5%.

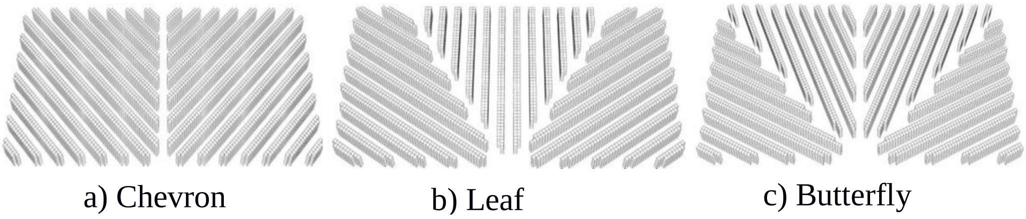



**Figure 1.3** Optimal Designs for Single-Command Operation. Adapted from: "Optimal unit-load warehouse designs for single-command operations." By Öztürkoğlu, Ö, Gue, K. R., & Meller, R. D. (2012). IIE Transactions, 44(6), 459-475.

Pick-up and deposit point (P&D) is a location in a warehouse that picker starts to collect products and deposit them in this location. There are studies in the literature where there are different numbers of P&D points in different places. In the study of Gue et al. (2012), P&D point is placed in front of warehouse and in Öztürkoğlu et al. (2014) P&D point is located at different locations of warehouse. According to Öztürkoğlu et al. (2014)'s study, multiple P&D point innovative designs decreased the average single-command travel distance between in the range of 3-10%. These designs are demonstrated in Figure 1.4.

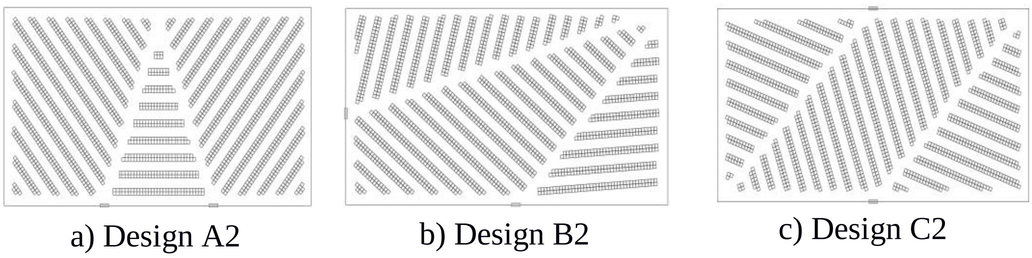



**Figure 1.4** Innovative Warehouse Designs for Multiple P&D points. Adapted from: "A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points." By Öztürkoğlu, Ö, Gue, K. R., & Meller, R. D. (2014). European Journal of Operational Research, 236(1), 382-394.

For sequential order picking operation, Dukic and Opetuk (2008) showed that traditional Design B is better than Fishbone design. However, Çelik and Süral (2013) showed that there are some cases where Fishbone design has 5-10% improvement less routing length than Design B.

### 1.1.2. Order Picking (OP)

One of the most important factors affecting average order picking length is to determine in which areas the products will be stored and stocked in. The most frequently cited strategies in industry and literature are randomized, dedicated, class-based and turnover-based product storage policies. Because of ensuring the usage of storage areas with high efficiency and simplicity of implementation, random storage strategy is used frequently in the industry. According to this method, products are randomly placed with the equal probability to the appropriate free storage cells (Petersen, 1999). Dedicated storage policy involves placing products in predefined areas for themselves. In this policy, because the placement of products in certain cells gains awareness to collectors, productivity increases in order picking. However, the policy causes low capacity utilization up to 50% at warehouses in which the seasonality is experienced or the rate of transformation is low (Bartholdi and Hackman, 2011). Class based storage policy has emerged by combining random and allocated field storage policies. In this policy, products are classified into classes according to their turnover rate and a storage area is allocated to each product class according to the total storage needs and proximity of the cells to the P&D point. Subsequently, products in each class are placed according to the random product placement policy in the respective cells allocated for its class. In traditional warehouse designs, there are three generally accepted applications of this policy; diagonal, across-aisle and within-aisle storage (Petersen and Schmenner, 1999).

The order picking process in the warehouses is the total tour of which the storage cells identified to be visited according to the given pick list. After visiting the last location in the tour, picker usually returns to the initial point. What is problematic is that the picker visits each cell only once and returns to the starting point. This is a special Travelling Salesman Problem (TSP) as called Steiner TSP which is most extensively studied in the literature and one of the NP-hard problem (Ratliff and Rosenthall, 1938; De Koster et al., 2007; Theys et al., 2010). The most common approach is these kinds of studies are assuming this sequential order picking route as TSP and modelling it. In order to solve these problems, exact algorithms and heuristics are chosen. In traditional layouts exact algorithms like linear programming, dynamic programming and branch-and-bound algorithms can calculate the optimal tour length. Ratliff and Rosenthal (1983)'s pioneering study showed that an optimal order-picking route can be solved in polynomial time using a dynamic programming-based exact algorithm for one-block warehouses. Roodbergen and De Koster (2001a) extended this algorithm for two-block warehouses, and developed an exact algorithm to solve the order-picker route in polynomial time. Gelders and Heeremans (1994) and Roodbergen and de Koster (2001b) solved the optimal order-picking route as a traveling salesman problem (TSP) by using Little et al. (1963)'s branch-and-bound algorithm. 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). Although exact algorithms are successful to finding optimal tour length for traditional layout, heuristic methods are more preferred to calculate total order picking tour length for non-traditional layouts. Greedy algorithms (Cormen et al., 1990) and constructive heuristics like Genetic algorithms (Eiben et al., 1994), Simulated annealing (Metropolis et al., 1953), Tabu search (Glover, 1986) etc. are some of the famous heuristic algorithms which applied in many studies which based on TSP. Specially metaheuristic algorithms are preferred in recent years to solve TSP based problem (Panda, 2018; Mazidi and Damghanijazi, 2017; Zhang et al., 2018). They have trade-off criteria; they can be shortened the calculation time but deviation from optimal of algorithm may be increased consequently.

There are limited studies in the literature for order picking operation in non-traditional warehouse layout. Dukic and Opetuk (2008) studied Fishbone layout and used an s-shape algorithm which is a special heuristic algorithm to compare efficiency with Design B. However, because they couldn’t find optimal total tour length, they couldn’t prove the improvement of Fishbone. Çelik and Süral (2014) calculated the optimal tour in Fishbone and showed that 5-10% less tour over Design B in small pick list sizes. Although using an exact algorithm to solve TSP gives more accurate solutions, it is very exhausting way to investigate a new non-traditional warehouse design problem. Because the main order-picking routing algorithms rely on a specific layout, they lack the flexibility to be used if the layout changes. Instead of applying exact algorithms, recently developed heuristic methods which perform good at finding close to optimal solution are more effortless.

# CHAPTER 2
# MODELLING THE WAREHOUSE

As mentioned in the previous section, there are many innovative design studies in the literature that have broken many design rules in different ways. (Gue and Meller, 2009; Öztürkoğlu et al., 2012; Öztürkoğlu et al., 2014). Although these designs provided reductions on travel distance, they cause extra storage area to because of the loss of locations due to angled and additional cross aisles. For example in Öztürkoğlu et al. (2014), the two non-intersecting aisle designs showed that expected travel distance can be reduced by 3-10% by needed 5-14% more space. The feasibility of a warehouse design is to offer a higher level of improvement than the area is needed. When suggesting a new layout, this performance should be taken into account.

The importance of the order picking operation is mentioned in the previous section. We can see that most of previous studies on this topic are about traditional layouts. However, there are limited studies on order picking tour in the non-traditional warehouses. Therefore, to fill the significant gap on a need of a warehouse design that reduced order picking tour length in the literature, in this thesis we examine a new warehouse layout design that can make order picking operation more efficient. In addition to innovative designs that have broken many design rules, designing and modelling of a warehouse which has only two intersecting aisles has been investigated in this thesis.

We have several design and model assumptions for our problem. The design assumptions are listed below.

- 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 centre of the warehouse. In particular, the presence of the

centre point is expected to facilitate accessing between the storage cells of the four different regions. They are also allowed to be originated from any side of the warehouse. Additionally, 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 2.1 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. $E_1$ and $E_2$ are referred as ending points of cross aisles respectively.

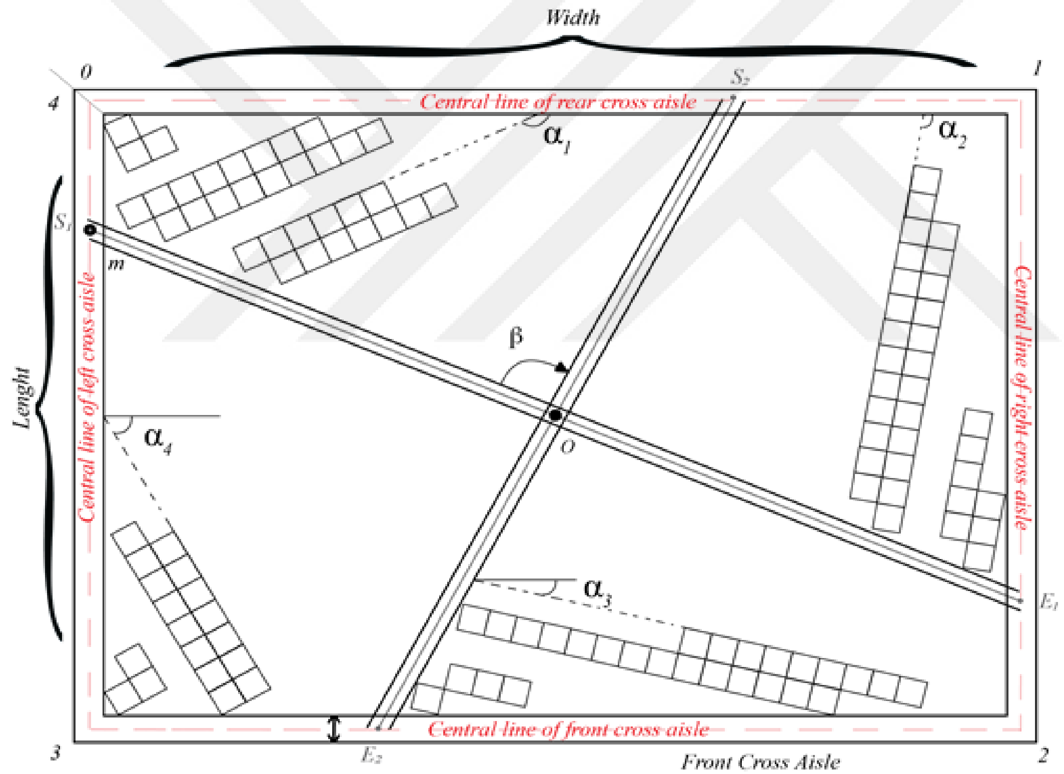



**Figure 2.1** X-shape Warehouse Model.

## 2.1. Encoding

The idea of representing a general warehouse design with a vector was seen first in Öztürkoğlu et al. (2014). With this vector, best warehouse layouts were searched and depicted. Based on the proposed encoding of Öztürkoğlu et al. (2014), we also develop our encoding to represent any warehouse design in our model. This encoding includes a set of continuous variables that are used to generate an X-shape warehouse layout: $\{S_{1x}, S_{1y}, S_{2x}, S_{2y}, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. 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 $i \in \{1,2,3,4\}$ are indicated by $\alpha_i$ ; $0 \leq \alpha_i \leq \pi$.

In order to reduce the number of variables, the rectangular shape warehouse is indicated by a continuous loop as suggested by Öztürkoğlu et al. (2014). Thus, the upper left, upper right, lower right and lower left corners take the values of 0, 1, 2, 3 respectively. 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. It can be easily converted to a point as long as the width and the length of the warehouse is given. Additionally, as assumed that two angled cross aisles are intersecting in the center of the warehouse, let $O_x$ and $O_y$ are the x and y coordinates of the centroid as $(O_x, O_y)$, respectively. Because of the assumption of linear angled cross aisles, the end point of the first angled cross aisle can be easily obtained by using $m$ and $O$. Moreover, let $\beta$ be the clockwise angle ($0 \leq \beta \leq \pi$) between the first and the second angled cross aisles. When $m$ and $\beta$ is known, the initial point of the second angled cross aisle can easily be obtained, as well as its end point. 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. As shown in Appendix 1, we can reduce the search space of $m \in [0,2]$.

After specifying the warehouse layout in our model, according to the specified product placement policy, the calculation of average picking tour length has been made. The representation vector and thus the design is altered for searching the best layout within each class.

### 2.2. Layout Generation

Traditional warehouse layouts are as we mentioned in section 1, most preferred layout types due to being well-balanced at two key indicators; easy to implement and acceptable picking tour length. It can be argued that non-traditional warehouse layout's appliance is not so effortless. Therefore, the main aim is developing a warehouse layout to create an environment for shorter order picking route under constraints to be preferred over the traditional layout. To compare to traditional warehouse and our warehouse model, there are 3 different width (W)/ length (L) ratios as warehouse size are selected. These ratios determine the capacity, height and width.

Our warehouse model has extra parameters different from Design B warehouse. Some of parameters are fixed for every shape ratio in one pallet unit (1 PU= 10 pixels). Such as;

- Cross aisles' width is decided as 2.0 PU.
- Pallet size is another fixed parameter which is 1.0 PU.
- The width of picking aisles "a" is fixed at 1.5 PU.

Some of parameters are changed for every different layout. In our non-traditional model there are two cross aisles and to identify their location $m$ and $\beta$ variables were used. As mentioned in "Encoding" part, picking aisles' angle $\alpha_i$ can be differ for every warehouse layout.

After deciding the parameters of the warehouse, next step is deciding which steps are going to follow for constructing a non-traditional warehouse layout. With shaping outer edges and the main cross aisles of the warehouse formed by the resolution of the vector, there are simple warehouse regions are developed. Then according to our assumption, for the picker to travel around the warehouse, side cross aisles are placed in the layout. The whole warehouse design is constructed by following the steps in the Figure 2.2.

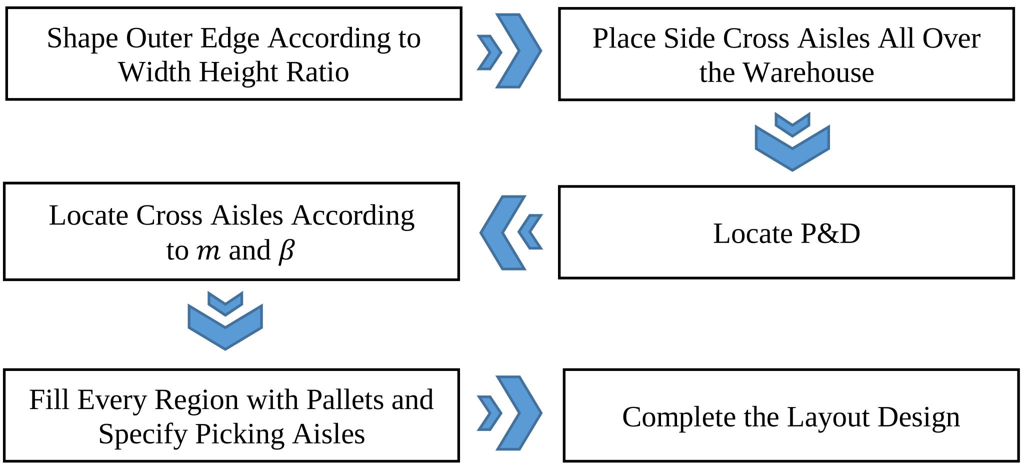



**Figure 2.2** General Steps for Layout Construction.

### 2.2.1. Region Construction

A region in a layout refers a separated storage area which is defined by the side and the angled cross aisles. As we mentioned before, the picking aisles in a region are willing to be parallel to each other. And all the picking aisles in the region have the same angle.

The most critical part in our module is to define each region appropriately. As seen in the Figure 2.3, there might exist regions with different shapes. These can be seen as triangular, quadrilateral or pentagonal shaped regions according to the orientations of the cross aisles. Therefore, to be able to construct these kind of shapes, their corners should be determined appropriately. After all the regions are described, to construct racks, we need to define a reference point for placing the initial pallet location.

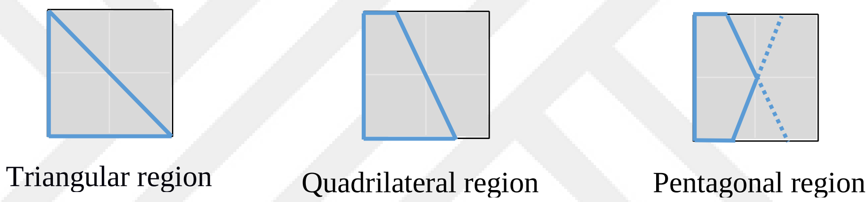



**Figure 2.3** Defining Regions.

### 2.2.2. Reference Point

The first and significant step is the process of determining the reference point. The reference point refers to a point where the first pallet in the storage area is placed. With placing pallets according to first pallet, picking aisles are formed which is explained in detail in next section. The incorrect selection of this reference point means the pallets and picking aisles in the entire warehouse area are designed incorrectly. Because of that, the instructions that are decided carefully must be followed.

As in the cartesian coordinate system, the reference point is determined at the plane where two forming lines are intersecting at point 0 while the x line increases to the right side and the y line increases downward. The simple logic is to find the point at which the greatest number of pallets are placed in the predetermined area. This investigation is done according to the angle of picking aisles. If there is only one reference point candidate in an area, that point is designated as the reference point.

But also an area can have more than one reference point candidate. If picking aisles' angle is $\alpha_i = 0$ or $\alpha_i = {}^{\pi}/_{2}$ and there is a corner point in the area, that corner point is selected as the reference point. If any of the top points are not a corner, then the most appropriate point at the edges is defined as the reference point. If $\alpha_i \neq 0$ or $\alpha_i \neq {}^{\pi}/_{2}$ then the steps of determination of the reference point in that case is shown as a flowchart at Figure 2.4. To guarantee the first pallet placement, the further intersection point is set as candidate reference point. Therefore, we determine two alternative reference points, chosen from the corners of the storage area. The reason is that, when we check the first pallet, if it is located out of the region, then we take the other reference point. In order to pick an appropriate reference point, we consider the angles of the picking aisles in the region. So that we create all parallel racks without any interruption starting from the reference point.

Draw a line passing through from centre with $\alpha_i$ (named as centre line)

Continue to draw lines parallel to the centre line until cover the whole determined area

Draw an extra parallel line farther from determined area (named as control

Calculate the nearest and furthest parallel line from the control line

Calculate the intersection points of the edges of the area with the nearest and furthest parallel lines

Set nearest intersection point as reference point

**Figure 2.4** Flowchart of Determination of Reference Point.

### 2.2.3. First Pallet

According to the nature of the research subject, the areas are formed between warehouse edges, related main cross aisles and intersection point of cross aisles. These areas should be filled with pallets. The first pallet refers to the initial pallet location in a region. A pallet location represented by its centroid and corners. Hence according to alternative reference points and the angle of the picking aisle in the region we create the centroid of the first pallet and then its corners. We check is it located in the region. If it is not, then we use the alternative reference point. In order to calculate the centroid of the first pallet as follows;

The pallet establishment is started with examination and calculation of the first pallet position according to the selected reference point in the area. The calculation is made with $\beta$ which is the angle between main cross aisles and $\alpha_i$ which is the angle between picking aisle and the warehouse edge. As a result of our work, 4 different main representations can cover all pallet versions for all possible $\alpha_i$ values. In Figure 2.5, these 4 main pallets are shown.

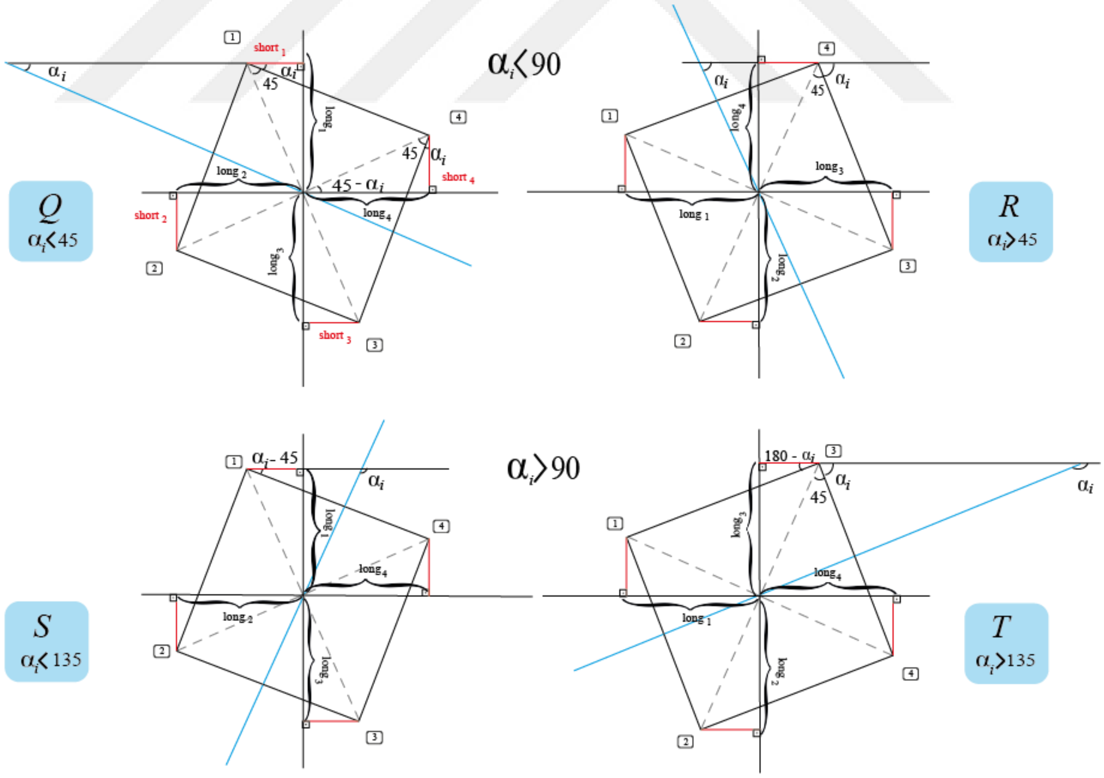



**Figure 2.5** Main Pallets.

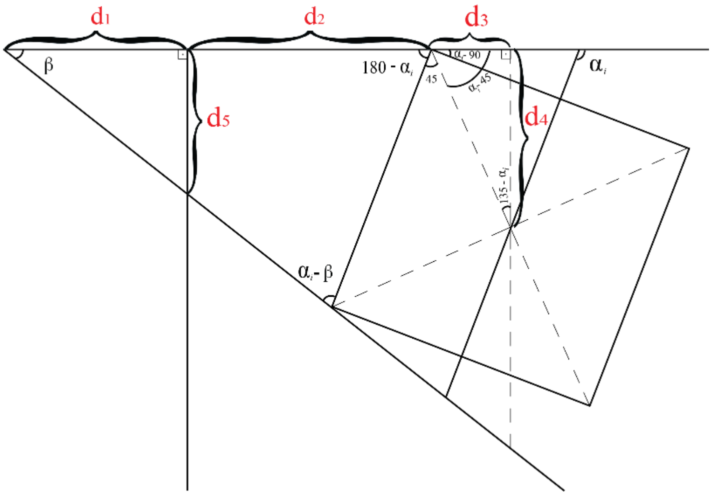



**Figure 2.6** Example of Calculation of Initial Pallet.

Corners and the centroid of initial pallet calculation is illustrated in Figure 2.5 and a detailed example of placing initial pallet in the warehouse shown in the Figure 2.6. The calculation is performed together with the equations given in the Table A2.1 in the Appendix 2 and the main pallets in the Figure 2.5.

### 2.2.4. Other Pallets

In order to build a rack, pallets should be continuing in the defined area. After the first pallet is placed, other pallets in the rack must continue to locate in accordance with the appropriate angle of picking aisles in the region. At step one, it is checked whether there is enough space for one pallet by following $\alpha_i$ angle adjacent to the initial pallet. After placing a pallet successfully, sometimes there will not be enough space to fit another pallet one on the orbit while forming up an aisle. In this case, overflow pallet is placed just below or above (according to the direction for construction of whole region) of the previously placed pallet. An example of this pallet placing policy is shown in Figure 2.7. The new pallet was tried to be located in the direction of the arrow (parallel to the angle $\alpha_i$) to the nearest (in this example to the right) of the genesis pallet which is indicated by darker colour on the upper left corner. Due to lack of place, unfortunately the pallet that has grey colour cannot be placed in defined area. So, we move on to build another rack. To do that, we place a pallet just below to genesis pallet. Because of having not enough space, instead of placing to the left side first and continuing with the right side; new pallets continue to be placed to the right. (Positioning was made, taking into account that the warehouse was viewed from above.) As presented in Figure 2.7 without extending the area, "two lines of pallet-picking aisle-two lines of pallet" pattern continue to construct whole area with following $\alpha_i$ angle. These steps progress until all four regions filled by pallets.

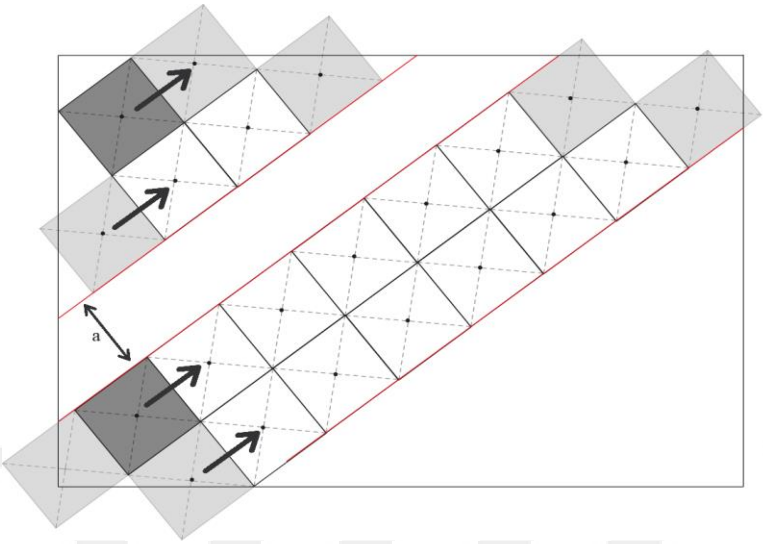



**Figure 2.7** Building Racks.

## 2.3. Network Construction

This stage aims to generate a network of nodes and edges that represent the pallet locations and intersects of aisles in a typical warehouse layout to ease of travel path calculations. There are similar approaches in the literature that converted a given a warehouse layout to a graph (Radliff and Rosenthal, 1983; Roodbergen, 2006; Öztürkoğlu, 2011; Ansari, 2017).

A typical warehouse network consists of a set of special nodes called access nodes, pick-up and deposit nodes, travel nodes and cross nodes. Access nodes are the points that are located on the central line of a picking aisle (see Figure 2.8). 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 centres of the pallet locations are used to calculate the coordinates of these access nodes. The calculation of access points' location is shown in the Figure 2.8. The first step of the calculation is determination of which pallet's access node is to be calculated. Then a line is drawn parallel to the warehouse edge from the center of the corresponding pallet. As a next step, a parallel line is drawn vertically to the center point of the aisle. When we connect center of pallet and middle point of the aisle, we get a right-angled triangle as can be seen in the Figure 2.8. Because of width of picking aisle and the $\alpha_i$ angle and the side cross aisle are already known, with a simple Pythagoras equation, the access points' coordinates are calculated. These steps are followed for locating all the access nodes in the warehouse.

The pick-up and deposit nodes represent the existing P&D points from and to where materials go through. In our model we assumed that it is fixed in the middle of front cross aisle. 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 (see Figure 2.8 for an example nodes). 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.

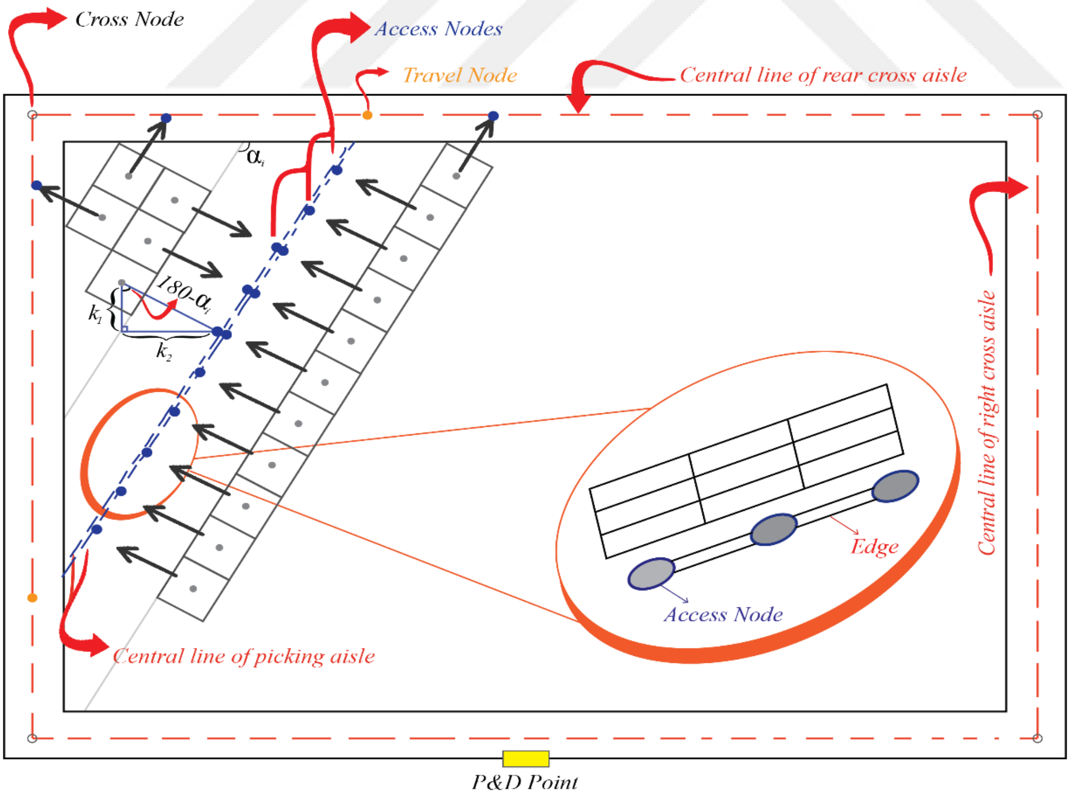



**Figure 2.8** Travel Nodes, Access Nodes and Cross Nodes Representation.

# CHAPTER 3
# ROUTE LENGTH CALCULATION

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. As last, this problem resembles to well-known TSP problem and the aim is to find the shortest route length for order picking operation.

As we mentioned in Section 1.1.2, there are many exact and time-efficient algorithms in the literature to solve TSP in traditional one-block or multi-block layouts (Ratliff and Rosenthall, 1938; De Koster et al., 2007; Theys et al., 2010). In most of these studies, the researchers focused on calculating optimal or close-optimal order-picking length in predefined traditional layouts like Design A and Design B. However, our problem has changeable layout. And when the layout is changed these algorithms cannot be easily adaptable. Because the aisle design problem in this study aims to find the best design parameters that minimize order-picking tour length, we decide to use one of the most widely used and TSP solving activity is proven metaheuristics algorithm: Ant Colony Optimisation (ACO).

## 3.1. Ant Colony Optimisation (ACO)

For our model's route length calculation, we preferred the Ant Colony Optimization (ACO) algorithm which is 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 way and the pheromone amount will increase so the shortest way has been found in this way.

It is showed as algorithm starts with $c$ number of ants placed randomly. At each step, the ant will probabilistically determine the next point, 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 $\tau_{ij}$ represents the pheromone amount between point $i$ and $j$. $\eta_{ij}$ is visibility intuitive value, which is the reverse of distance between $i$ and $j$ point ($1/_{d_{ij}}$). $PR$ 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.

After all the ants complete visiting the all orders we get total tour length. In order to improve the solution 2-opt shifting local search algorithm is implemented. This algorithm is for shifting the two consecutive links to search better solutions. We prefer this algorithm because it is easy to implement and efficient way to improve the solution. In our problem, we check the tour length that calculated after shifting operation whether is less than previous tour, or not. If it is improved, then we update the solution and tour length. In case there are no improvement, the ACO algorithm continues with current non-shifted solution. The pseudocode of 2-opt is given in Figure 3.1. The used algorithm represented in a flow chart referred as Figure A4.1 in the appendix.

**Algorithm 2:** 2-Opt Algorithm

Let $i$ and $j$ are nodes, $(i + j = n)$, $\quad i, j, n \in N^+$
Let $d_{ij}$ represents distance between $i$ and $j$ nodes, $\quad d_{ij} \in R$

**For** $\{i = 0,\ i < n - 3,\ i++\}$
  **For** $\{j = i + 2,\ j < n - 1,\ j++\}$
    **If** $\left(d_{ij} < d_{i(i+1)}\right)$ AND **If** $\left(d_{(i+1)(j+1)} < d_{i(i+1)} + d_{j(j+1)}\right)$
      Swap $i$ and $j$,
      Calculate the new total route length
    **Otherwise**
      Do not swap

**Figure 3.1** Pseudocode of 2-Opt Algorithm.

We chose examples of problems with the number of cities because pick list sizes of our problem are varieties 5 to 50. Performance of the developed algorithm has been tested using examples of problems given in TSPLIB (Reinelt, 1991). The sizes of example problems are selected close to predetermined pick list size. The values of the parameters used in the ACO algorithm are as follows; stopping criteria is 100 iterations, $c$ is 30, $ER$ is 0,75 and finally $PR$ is 1 and $VR$ is 2. In addition, the algorithm was run with 10 different random number seed. According to the results (shown in the Table 3.1) shows that the developed ACO algorithm has capability of being able to use for calculating average order picking tour length. There is a slight deviation from optimal for 50 pick list size but for our problem we think that this algorithm is applicable. To be able to run the ACO to calculate shortest tour length in our warehouse, one of the need is the shortest distance matrix for the points that need to be visited which is given in the ACO equations as $d_{ij}$.

**Table 3.1** ACO Algorithm Results.

| Example Problem | Opt. Value | Min | Avg. | Max | CPU Time (sec) | Deviation from Avg. Opt. |
|---|---|---|---|---|---|---|
| **uly16** | 6859 | 6859 | 6859 | 6859 | 0.2 | 0.00% |
| **gr17** | 2085 | 2085 | 2085 | 2085 | 0.2 | 0.00% |
| **uly22** | 7013 | 7013 | 7015 | 7036 | 0.4 | 0.03% |
| **fri26** | 937 | 937 | 937 | 937 | 0.5 | 0.00% |
| **dan42** | 699 | 699 | 699 | 699 | 1.1 | 0.00% |
| **eil51** | 426 | 428 | 434 | 438 | 1.7 | 1.97% |

### 3.2. Distance Matrix

In order to calculate total tour length, we need to calculate shortest paths between points to visit. There might be a several paths between two nodes, naturally. The shortest path algorithm of Dijkstra is used to solve such problems which is a one to many algorithm (Dijkstra, 1959). For $n$ number of point, Dijkstra algorithm runs for $n \cdot n$ matrixes with cost of $O(n + n \log n)$. Normally according to pick list size, distance between all access nodes and P&D nodes and distance between access nodes should be calculated with Dijkstra algorithm. However due to its cost, there is no need to use this algorithm for all nodes to build a distance matrix. We developed a new algorithm for calculating distance for the required locations which are referred by access nodes. This algorithm basically performs the following; firstly, we need to create a main distance matrix using Dijkstra which includes the distances between the cross points and the travel points. Then we need to compute the distances between access nodes using main distance matrix. There are two alternatives to calculate shortest path between two access nodes according to their locations. First alternative; If both of them are aligned at the same aisle, the shortest path is calculated directly as a distance. Second alternative; If the two nodes are in separate aisles, different paths between these nodes are calculated with using main distance and then the shortest one is selected as the distance. The details of the algorithm are given in the pseudocode in the Appendix 3.

## 3.3. Design Optimisation: Differential Evolution Algorithm (DE)

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 Algorithm (DE).

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 3.2. With the first step as initialization, 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 $X_{i,G} = \left[x_{i,G}^{1}, x_{i,G}^{2}, \dots, x_{i,G}^{D}\right]\ i = 1,2,\dots,P$. 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.

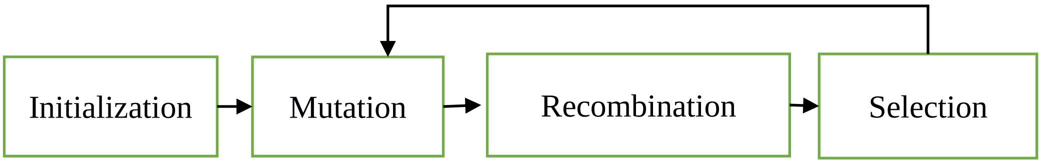



**Figure 3.2.**Main Steps of DE Algorithm.

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 $V_{i,G+1} = X_{r1,G} + F(X_{r2G} - X_{r3,G})$ formulation, a mutation vector is created. In this formulation, F is a mutation factor and $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, \ldots, v_{i,G}^D\}$ vector represents the donor vector.

In order to optimise our design problem in the mutation step, 4 different strategies were integrated in DE. The strategies given the formulas below were used together in equal probability. 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/Best/2: $V_{i,G} = X_{best,G} + F\left(X_{r_1^i,G} - X_{r_2^i,G}\right) + F\left(X_{r_3^i,G} - X_{r_4^i,G}\right)$ (Storn, 1996).
- DE/Rand/1: $V_{i,G} = X_{r_1^i,G} + F\left(X_{r_2^i,G} - X_{r_3^i,G}\right)$ (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 ($U_{i,G} = \{u_{i,G}^1, u_{i,G}^2, \ldots, u_{i,G}^D\}$) 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 $X_{i,G+1} = \begin{cases} U_{i,G}, & if\ f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & otherwise \end{cases}$, $i = 1,2,\dots,P$, 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 stop criteria as illustrated in the Figure 3.2.

### 3.3.1. Constraint Handling

As we explained in the section 2.1, there are six variables of our problem. $m$ is bounded with $[0,2]$, $\beta$ is defined as in the range of $[0,\pi]$ and $\alpha_i$'s can be change in the range of $[0,\pi]$ for every $i \in \{1,2,3,4\}$. While searching the best value in the DE algorithm, there is a possibility of these variables are out of bounds during mutation. In order to keep them in the boundaries, we use the Periodic Approach which is proposed by Padhye et al. (2015) as a constraint-handling strategy. This strategy, we bound 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; $x = \{m, \beta, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$.

### 3.3.2. Implementation of DE

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. Because according to them, in P=50 and in order to balance efficiency and speed, CR should be taken in the range [0.1-0.9] and F should be taken in the range [0.4-0.9] in steps of 0.1. As mentioned Mallipeddi et al.'s (2011) study, to balancing efficiency and speed in our problem parameters as P = 50, CR=0.5, F=0.4 implemented.

Calculations have been analysed on an example problem to understand the efficiency of DE. With this purpose, a continuous variable test function known as the "Six-hump Camel-back" in the literature is selected (Pohlheim, 2007). We chose this problem because it is similar the way our problem has continuous variables. The algorithm tested with 20 different seeds and showed an optimal deviation of 0.09% on the average. According to Thomsen (2004), different types of DE algorithm can find optimum points in single results, although in this test algorithm DE couldn't find the optimum, it shows close results which is an acceptable threshold for our problem.

In order to stop the algorithm, we needed a criterion. We use iteration number to stop algorithm but also 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 results are controlled by checking whether they are better than the previous results. If there is not, algorithm stops at iteration 1100. However there is an improvement, the algorithm runs 100 iterations more than the iteration that seen improvement. Also, the improved result set as global solution. The algorithm continues to run until there is no better solution found in next 100 iterations.

# CHAPTER 4
# RESULTS AND CONCLUSIONS

## 4.1. Numerical Study

This section describes the search for the best X-shape warehouse layout where average tour length is minimized. As mentioned in 2.2, we study 3 different shapes of rectangular warehouse that has storage racks placed in a single-deep rack 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 verifying 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 location 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 $n$, $\mathrm{C}(m, n)$ 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 (5) pick list size and for 3 different shapes of warehouse. In the Table 4.1, averages and standard deviations of these orders are given. According to statistical analyse for finding sample size with $n = \frac{s^2 \cdot Z^2_{\alpha/2}}{(\bar{x} \cdot 0.01)^2}$ formula we calculate our sample sizes for number of orders in 95% confidence interval and %1 margin of error. As a result, we decided to use 1500 samples for smaller pick list size (3,5,10) and 250 samples for larger pick list size (20,30,40,50) is sufficient.

**Table 4.1** Pick List Analyses.

| Warehouse Size | 500x500 | | 750x750 | | 1000x1000 | |
|---|---|---|---|---|---|---|
| # of Pick | 3 | 50 | 3 | 50 | 3 | 50 |
| E[0]= | 1204,65 | 3339,53 | 1613,79 | 4751,389 | 1618,78 | 7067,86 |
| E[1]= | 1242,93 | 3388,65 | 1882,85 | 5467,94 | 1800,92 | 7528,55 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| E[998]= | 1254,10 | 3291,99 | 2246,44 | 4516,01 | 2329,89 | 7319,08 |
| E[999]= | 1251,44 | 3525,16 | 2111,60 | 5577,22 | 2201,37 | 7073,10 |
| $\bar{x}$ | 1300,18 | 3348,75 | 1910,15 | 5200,41 | 2516,56 | 7063,79 |
| S | 217,12 | 223,16 | 366,33 | 356,66 | 496,16 | 492,76 |
| n | 1071,29 | 170,59 | 1412,92 | 180,69 | 1493,28 | 186,93 |

Thus, the average tour length of these order list sizes (the average tour length) determine the fitness value or cost of the design concerned. With ACO algorithm, we try to minimize fitness value in 3 different shape ratios as mentioned in Section 3.1. In order to compare efficiency of generated non-traditional warehouse, we made Design B as a base for deciding the capacity which are represented at Table 4.2. We use the same order lists and very close storage capacity for both X-shaped and Design B to make an accurate comparison.

**Table 4.2** Design B Warehouse Features.

| Shape Ratio | Aisle Number | Width (PU) | Length (PU) | Total Pallet |
|---|---|---|---|---|
| 1:1 | 7 | 380.0 | 420.0 | 510 |
| 2:1 | 11 | 580.0 | 290.0 | 512 |
| 3:1 | 14 | 730.0 | 240.0 | 510 |

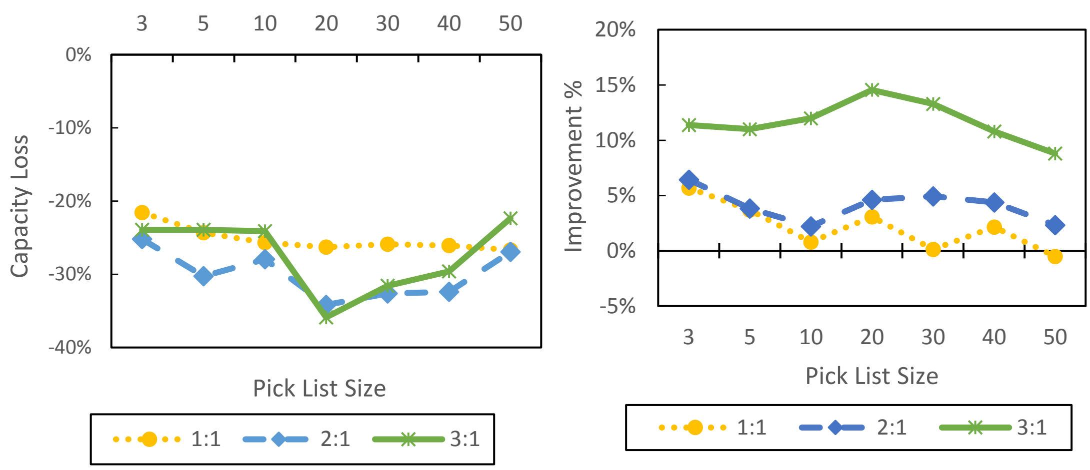
## 4.2. Results

We run our experiments with three different seeds on a computer running on 4 GB RAM and a 1.70 GHz Intel ® CORE i5 processor and select the one with the best result. The average computational time for these experiments in three different shape ratios are shown at Table 4.3. In the previous section we mentioned that the creation of picking list for optimizing the warehouse design and the calculation of average tour length could be takes very long time. Table 4.3 shows that even the smallest CPU time which is the warehouse in 2:1 shape ration for 3 picks takes 4 hours.

**Table 4.3** Average Running Times.

| Pick No | | 3 | 5 | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|
| Avg CPU Time (hr) | 1:1 | 5,15 | 11,09 | 39,57 | 13,43 | 39,40 | 70,82 | 114,39 |
| | 2:1 | 3,85 | 9,14 | 21,86 | 23,79 | 55,97 | 84,22 | 116,75 |
| | 3:1 | 4,35 | 10,09 | 40,19 | 86,82 | 34,55 | 85,33 | 110,22 |

There are representations of the best X-shape layouts for all picking number of orders in the Appendix 5. In Table A6.1 shown the tour lengths and specifications of these layouts of proposed non-traditional layout (X-shape) designs which has equal dimensions with equivalent traditional warehouse design. Also, average tour lengths of 2-blocked traditional layout (Design B) for the same circumstances are presented in the same table. Although there is significant improvement of proposed warehouse layout over Design B (see Figure 4.1.b), as seen in the Figure 4.1.a the capacities of X-shape layouts are much lower than same shaped Design B, which is not a healthy analogy.

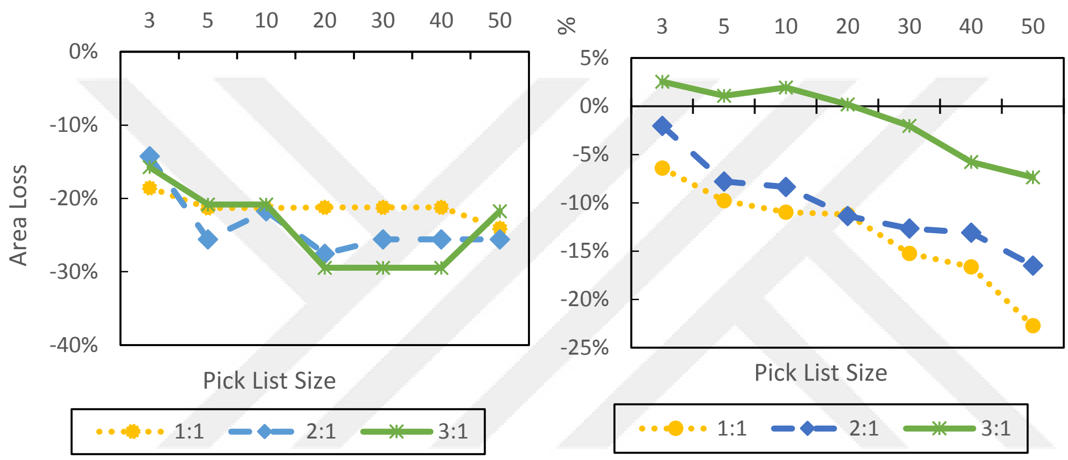



a) Capacity Loss over Design B

b) Improvement over Design B

**Figure 4.1** Comparison of X-shape and Design B in Same Size.

In order to make a correct comparison, X-shape warehouse dimensions are expanded until their capacities' approximately equal to equivalent Design B. While expanding warehouses, obtained variables from optimization (such as $\alpha_i$'s and m) and shape ratio are kept constant. In this case, it turns out new X-shape layout needs extra space. In order to visualize this need, Figure 4.2.a represents the area loss over Design B. Expanded X-shape layout's specifications and average tour lengths (the fitness) are represented in Table A6.2**.** The improvements of the expanded X-shape layout over equivalent Design B which are shown in the Figure 4.2.b. For the warehouse with 1:1 and 2:1 shape ratio, X-shape layout has worse fitness values than Design B. However, X-shape warehouse that has 3:1 shape ratio in small amount of order picking, it is superior over traditional Design B. Purposed layouts for 3:1 shape ratio shows improvement at avg. tour length on the average 1,44% if there is 21,72% more area.




a) Area Loss over Design B    b) Improvement over Design B

**Figure 4.2** Comparison of Expended X-shape and Design B in Same Capacity.

If we interpret the solutions in Appendix 5, the cross aisles in model of 3:1 and 2:1 shape ratio for 3 orders looking alike traditional layout. With these cross aisles, it is easier to travel front and rear cross aisles in the warehouse. As seen in those layouts, the angles of picking aisles' which similar to Chevron design (Öztürkoğlu et al., 2012), is meeting the need occurs with separation of upper and lower part of warehouse with main aisles.

## 4.3. Conclusion

In this thesis, we investigate a completely new non-traditional warehouse design that will improve the order picking operation. We assume there are two cross aisles that intersect in the middle of rectangular shape warehouse under randomized storage policy and one P&D point which is located on middle of front cross aisles. Because cross aisles can be located every edge in the warehouse and picking aisles may have all kinds of angles in the range $[0, \pi]$, to optimize them we use Differential Evolution Algorithm. The fitness of our proposed warehouse is calculated with the Ant Colony Optimisation Algorithm and to compare our proposed warehouse layout efficiency, we select Design B layout as a base.

There wasn’t many research that non-traditional warehouse proposed for order picking operation. We can say that; this thesis is a gain for literature as a non-traditional warehouse design. However, results show that X-shape non-traditional warehouse is not the best alternative compare to traditional warehouses under random storage policy. With same dimension warehouse as Design B, X-shape shows improvement in route length on the average 5.9%, but also loses capacity on the average %27 because of with cross aisles, the proposed layout needs more area to balance the capacity. Even though there is a slight improvement for total tour length in warehouse that has 3:1 shape ratio with equal capacity, this layout needs more area than traditional one.

In light of results, the proposed warehouse is not a good version of Design B under these assumptions. If a long warehouse is to be constructed (like 3:1 shape) and there are no space concerns, we can recommend our layouts developed for small pick list size to sector managers. On the other hand, X-shape designs are not preferable for other shape ratio. Of course, there is a chance to get improvement for changing assumptions. So for future work, this idea can evolve under different acceptances. Different warehouse operation like single command or different P&D location, developments can be seen over traditional layouts.

## REFERENCES


Ansari, M., & Smith, J. S. (2017). Warehouse Operations Data Structure (WODS): A data structure developed for warehouse operations modelling. *Computers & Industrial Engineering*, 112, 11-19.

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

Berry, J. R. (1968). Elements of warehouse layout. *The International Journal of Production Research,* 7(2), 105-121.

Chackelson, C., Errasti, A., Cipres, D., & Álvarez, M. J. (2011, September). Improving picking productivity by redesigning storage policy aided by simulations tools. In *V international conference on industrial engineering and industrial management* (pp. 306-313).

Cormen, T. (1990). Charles Leiserson. Ronald Rivest. *Introduction to algorithms*, 485-488.

Çelik, M., & Süral, H. (2014). Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE transactions*, 46(3), 283-300.

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

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1), 269-271.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1), 53-66.

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

Eiben, A. E., Raue, P. E., & Ruttkay, Z. (1994, October). Genetic algorithms with multi-parent recombination. *In International Conference on Parallel Problem Solving from Nature* (pp. 78-87). Springer, Berlin, Heidelberg.

Farber, M. (2016). Consumers Are Now Doing Most of Their Shopping Online. Retrieved from http://fortune.com/2016/06/08/online-shopping-increases/

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

Gelders, L., & Heeremans, D. (1994). Het travelng salesman probleem toegepast op order picking. *Tijdschrift voor economie en management*, 39(4).

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research,* 13(5), 533-549.

Gue, K. R., & Meller, R. D. (2009). Aisle configurations for unit-load warehouses. *IIE Transactions*, *41*(3), 171-182.

Gue, K. R., Ivanović, G., & Meller, R. D. (2012). A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review,* 48(4), 795-806.

Hall, R. W. (1993). Distance approximations for routing manual pickers in a warehouse. *IIE transactions*, 25(4), 76-87.

Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research*, 222(3), 484-494.

Hong, S., Johnson, A. L., & Peters, B. A. (2012). Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research,* 221(3), 557-570.

Jitkongchuen, D., & Thammano, A. (2014). A self-adaptive differential evolution algorithm for continuous optimization problems. *Artificial Life and Robotics*, 19(2), 201-208.

Kitayama, S., Arakawa, M., & Yamazaki, K. (2011). Differential evolution as the global optimization technique and its application to structural optimization. *Applied Soft Computing*, 11(4), 3792-3803.

Kunder, R., & Gudehus, T. (1975). Mittlere wegzeiten beim eindimensionalen kommissionieren. *Zeitschrift für Operations Research*, 19(2), B53-B72.

Little, J. D., Murty, K. G., Sweeney, D. W., Karel, C., (1963). An algorithm for the traveling salesman problem. *Operations research 11* (6), 972-989.

Mallipeddi, R., Suganthan, P. N., Pan, Q. K., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied soft computing*, 11(2), 1679-1696.

Mazidi, A., & Damghanijazi, E. (2017). Meta-Heuristic Approaches for Solving Travelling Salesman Problem. *International Journal of Advanced Research in Computer Science*, 8(5).

Mesa, A. (2016). *A Methodology to Incorporate Multiple Cross Aisles in a Non-Traditional Warehouse Layout* (Doctoral dissertation, Ohio University).

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal* *of chemical physics,* 21(6), 1087-1092.

Oztukoglu, O. (2011). *New warehouse designs: angled aisles and their effects on travel distance* (Doctoral dissertation, Auburn University).

Öztürkoğlu, Ö., Gue, K. R., & Meller, R. D. (2012). Optimal unit-load warehouse designs for single-command operations. *IIE Transactions*, 44(6), 459-475.

Öztürkoğlu, Ö., Gue, K. R., & Meller, R. D. (2014). A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points. *European Journal of Operational Research*, 236(1), 382-394.

Panda, M. (2018). Performance Comparison of Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing Applied to TSP. *International Journal of Applied Engineering Research*, 13(9), 6808-6816.

Padhye, N., Mittal, P., & Deb, K. (2015). Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Computational Optimization and Applications*, 62(3), 851-890.

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*, 17(11), 1098-1111.

Petersen, C. G. (1999). The impact of routing and storage policies on warehouse efficiency. *International Journal of Operations & Production Management*, 19(10), 1053-1064.

Petersen, C. G., & Schmenner, R. W. (1999). An evaluation of routing and volume‐based storage policies in an order picking operation. *Decision Sciences*, 30(2), 481-501.

Pohl, L. M., Meller, R. D., & Gue, K. R. (2009b). Optimizing fishbone aisles for dual-command operations in a warehouse. *Naval Research Logistics (NRL),* 56(5), 389-403.

Pohl, L. M., Meller, R. D., & Gue, K. R. (2011). Turnover-based storage in non-traditional unit-load warehouse designs. *IIE Transactions*, 43(10), 703-720.

Pohlheim, H. (2007). Examples of objective functions. Retrieved, 4(10), 2012.

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation,* 13(2), 398-417.

Ratliff, H. D., & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, 31(3), 507-521.

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA journal on* *computing*, 3(4), 376-384.

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

Roodbergen, K. J., & Koster, R. (2001b). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research,* 39(9), 1865-1883.

Roodbergen, K. J., & Vis, I. F. (2006). A model for warehouse layout. *IIE transactions*, 38(10), 799-811.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G. J., Mantel, R. J., & Zijm, W. H. (2000). Warehouse design and control: *Framework and literature review. European Journal of Operational Research,* 122(3), 515-533.

Storn, R. (1996). 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.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.

Theys, C., Bräysy, O., Dullaert, W., & Raa, B. (2010). Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, 200(3), 755-763.

Thomsen, R. (2004). Multimodal optimization using crowding-based differential evolution. *In Evolutionary Computation, 2004. CEC2004. Congress on* (Vol. 2, pp. 1382-1389). IEEE.

White, J. A. (1972). Optimum design of warehouses having radial aisles1. *AIIE Transactions*, 4(4), 333-336.

Zhang, T., Ke, L., Li, J., Li, J., Huang, J., & Li, Z. (2018). Metaheuristics for the tabu clustered traveling salesman problem. *Computers & Operations Research*, 89, 1-12.

# APPENDIX 1 – SYMMETRICAL CASES

In Figure A1.1, there are 20 different scenarios for the variable $m$ where $m \in [0,2]$. In every four class (A, B, C, D), $m$ is fixed at bounds that written above them and $\beta$ angle is increased as shown in the figures.

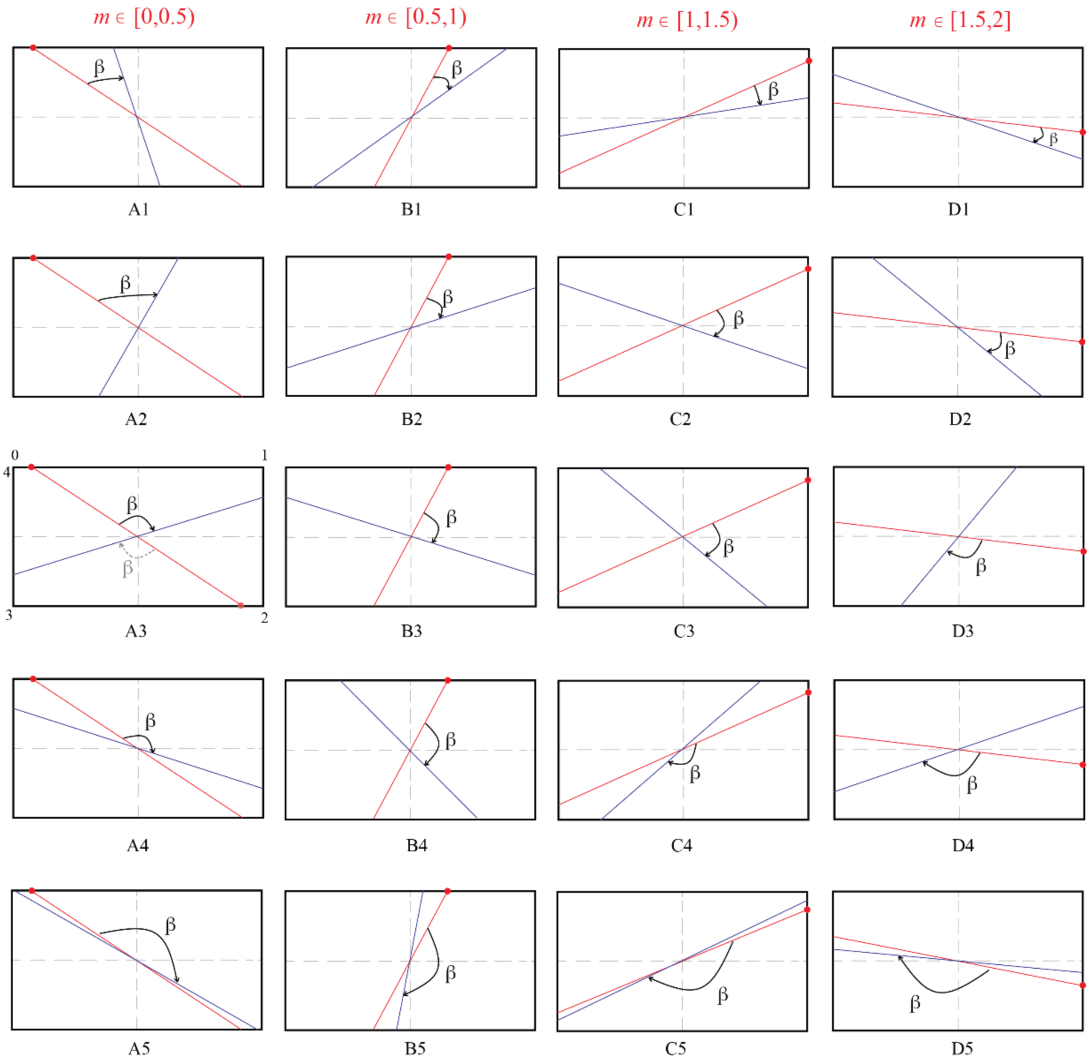



**Figure A1.1** Scenarios of $m$ in [0,2].

In the section 2.1, there is an argument for the restriction of $m$. To prove that; if we add 2 to $m$ in A3, $m$ will be in [2, 2.5]. With the same $\beta$ angle, we get exactly the same layout as A3. In addition, some symmetric and identical conditions have been identified from these warehouse design representations. For example; A3 and C3 are identical & A3 and B3 are symmetric to $x$ axis. Thus, as it can be seen all third representations in all classes are the same warehouse layout.

# APPENDIX 2 – FIRST PALLET REPLACEMENT

In this table of equations, one edge of each pallet is denoted by $p$ and pallet corners are denoted by *k*.

**Table A2.1** Equations for Construction of Pallet.

| | *Representative Pallet Type* | | *long* $_k$ | *short* $_k$ |
|---|---|---|---|---|
| $\alpha_i < 90^{\circ}$ | (Q) $\alpha_i < 45^{\circ}$ | $k$=1 | $\frac{p\sqrt{2}}{2} \times \sin(45+\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45+\propto)$ |
| | | $k$=2 | $\frac{p\sqrt{2}}{2} \times \sin(45+\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45+\propto)$ |
| | | $k$=3 | $-\frac{p\sqrt{2}}{2} \times \sin(45+\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45+\propto)$ |
| | | $k$=4 | $-\frac{p\sqrt{2}}{2} \times \sin(45+\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45+\propto)$ |
| | (R) $\alpha_i > 45^{\circ}$ | $k$=1 | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ |
| | | $k$=2 | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ |
| | | $k$=3 | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ |
| | | $k$=4 | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ |

| | *Representative Pallet Type* | | *long* $_k$ | *short* $_k$ |
|---|---|---|---|---|
| $\alpha_i > 90$ ° | (S) $\alpha_i < 135$ ° | $k$=1 | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=2 | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=3 | $\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=4 | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | (T) $\alpha_i > 135$ ° | $k$=1 | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=2 | $\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $-\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=3 | $-\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |
| | | $k$=4 | $\frac{p\sqrt{2}}{2} \times \sin(45-\propto)$ | $\frac{p\sqrt{2}}{2} \times \cos(45-\propto)$ |

# APPENDIX 3 – PSEUDOCODE OF SPEED-UP ROUTE LENGTH CALCULATION ALGORITHM

**Algorithm 1:** Speed-up Route Length Calculation Algorithm

Let $L$ be a list of required $n$ locations and $m$ P&D points.

$L = \{P_1, P_2, \dots, P_n, P_{n+1}, \dots, P_{n+m}\}$ where the first n point indicates requested locations, the last $m$ points are the P&D points

A

**For** $\{i = 1, i \leq |L|, i++\}$

**For** $\{j = 1, j < |L|, j++\}$

**If** $P_i$ and $P_j$ are picking locations

**If** locations $P_i$ and $P_j$ are on the same aisle;

$$d_{ij} = \left|P_{i_x} - P_{j_x}\right| + \left|P_{i_y} - P_{j_y}\right|$$

**If** locations $P_i$ and $P_j$ are on the different aisle;

Let $P_{i1}$ and $P_{i2}$ are the respective travel nodes (intersects) at aisle $i$,

$P_{j1}$ and $P_{j2}$ are the respective travel nodes (intersects) at aisle $j$.

$K\epsilon\{P_{i1}, P_{i2}\}$ and $R\epsilon\{P_{j1}, P_{j2}\}$, for all points of $\{k, r\}$

$k = 1,\ k \leq |K|;\ r = 1,\ r \leq |R|$

$$d_{kr} = \sqrt{\left(P_{i_x} - P_{ik_x}\right)^2 + \left(P_{i_y} - P_{ik_y}\right)^2} + \sqrt{\left(P_{j_x} - P_{jr_x}\right)^2 + \left(P_{j_y} - P_{jr_y}\right)^2} + D^T\left(P_{ik} - P_{jr}\right)$$

$d$ is the minimum of $d_{kr}$

A

Let $T$ be a set of locations including all $m$ P&D points and all $k$ travel nodes.

$$T = \{\underbrace{t_1, t_2, \dots, t_m}_{\text{P\&D Points}}, \underbrace{t_{m+1}, \dots, t_k}_{\text{Travel Nodes}}\}$$

$$D^T = \begin{bmatrix} 0 & t_1 t_2 & \cdots & t_1 t_k \\ t_2 t_1 & 0 & \cdots & \cdots \\ \cdots & \cdots & 0 & t_{k-1} t_k \\ t_k t_1 & \cdots & t_k t_{k-1} & 0 \end{bmatrix}$$

**For** $\{i = 1, i \leq |T|, i++\}$

**For** $\{j = 1, j < |T|, j++\}$

**If** $t_i$ and $t_j$ are P&D points

$$D^t(t_i t_j) = |t_{i_x} - t_{j_x}| + |t_{i_y} - t_{j_y}|$$

**If** $t_i$ and $t_j$ are travel nodes but they belong to same aisle (they are the intersection points of same aisle)

**Else**

Use Dijkstra Algorithm

# APPENDIX 4 – ANT COLONY OPTIMISATION

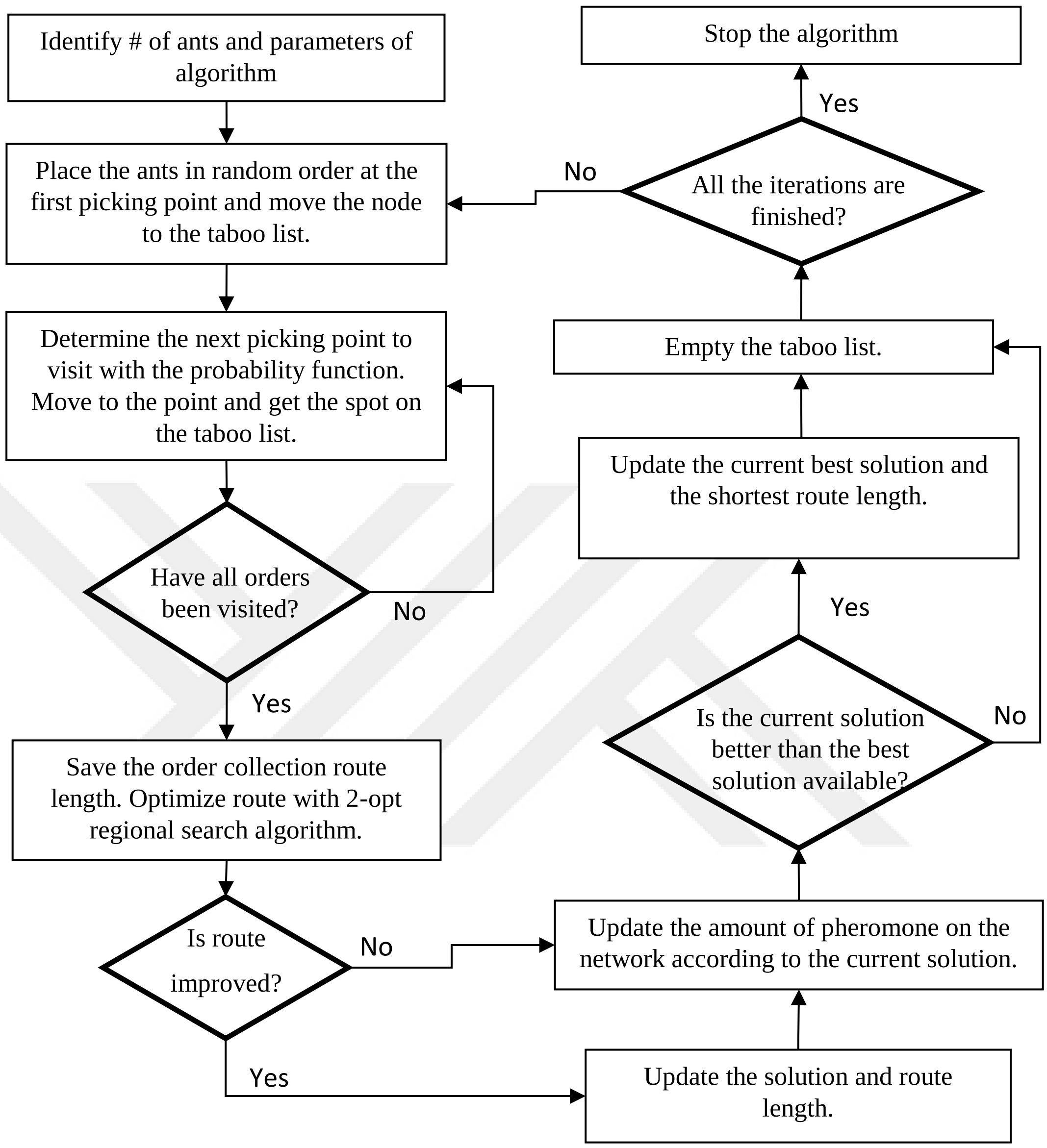
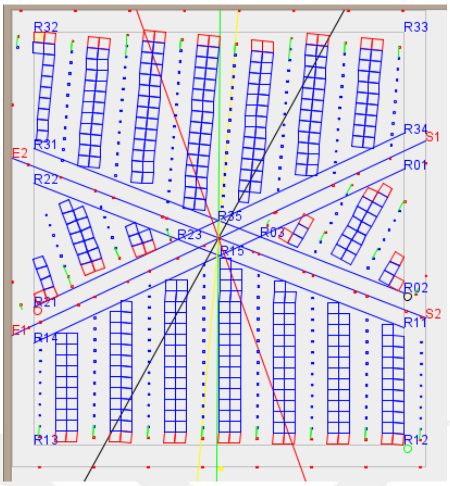
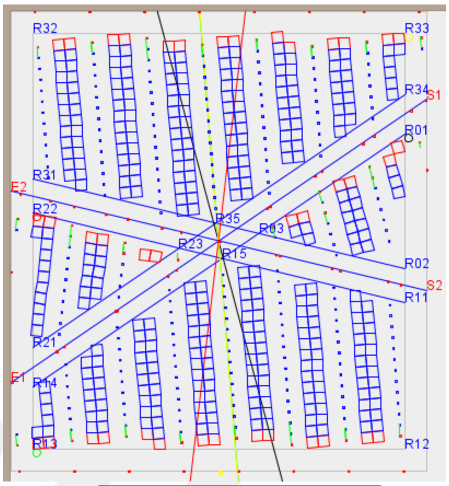



**Figure A4.1** Flowchart of Ant Colony Optimization Algorithm.

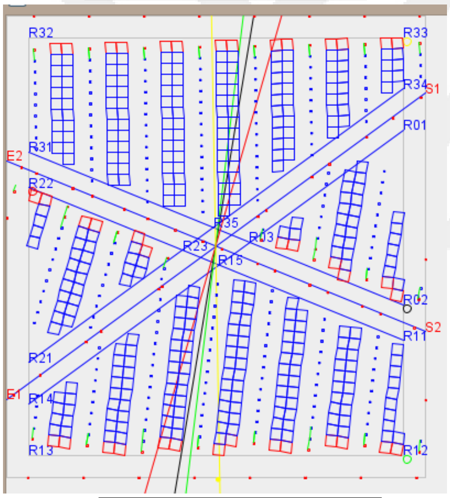
# APPENDIX 5 – X-SHAPE WAREHOUSE DESIGNS



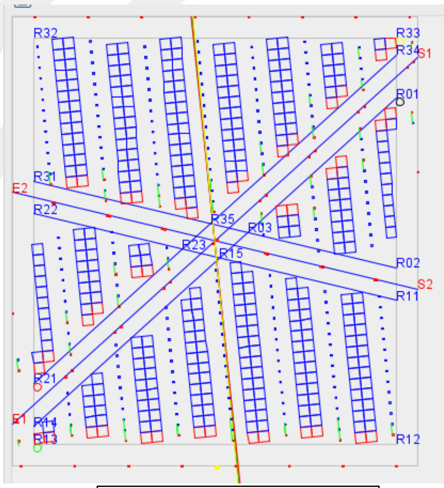

For 3 pick



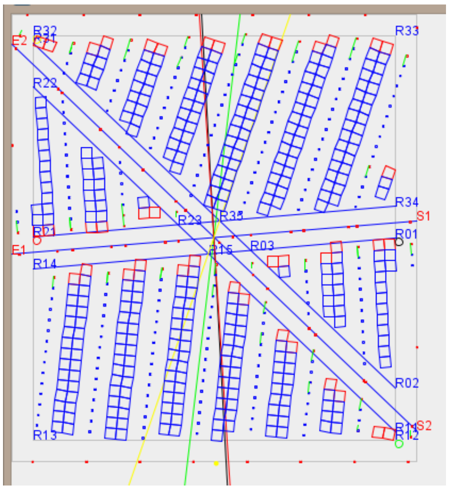

For 5 pick



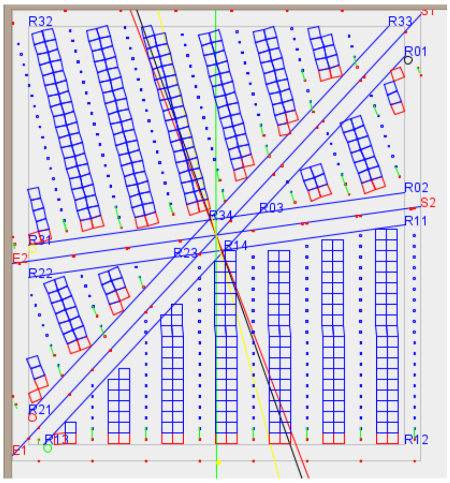

For 10 pick



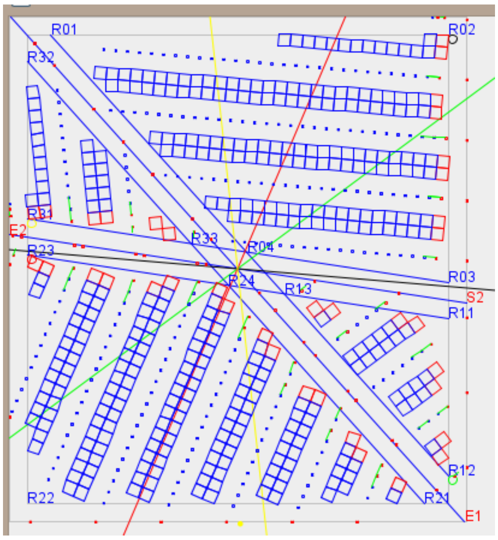

For 20 pick




For 30 pick



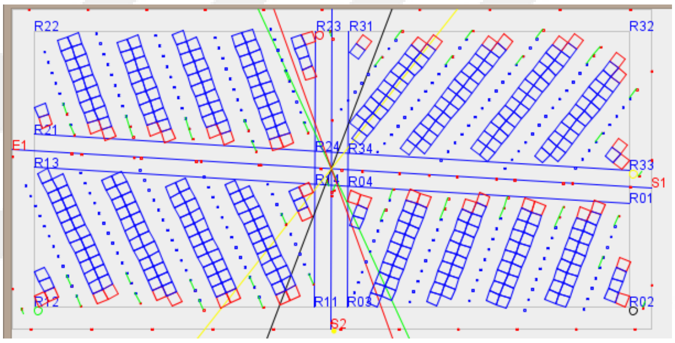

For 40 pick



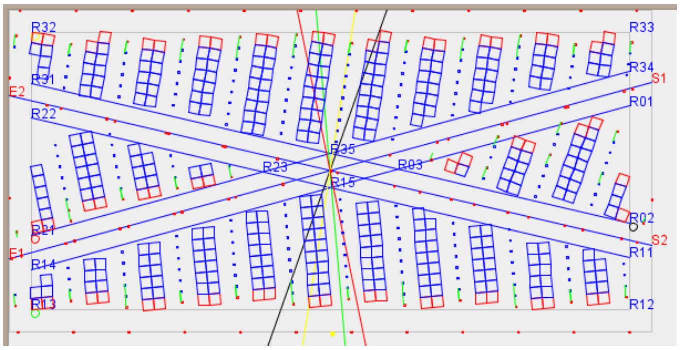
For 50 pick

**Figure A5.1** Warehouse Representations of 1:1 Shape Ratio.

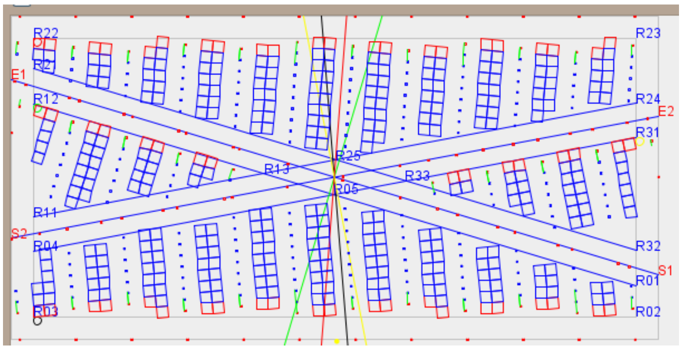
For 3 pick



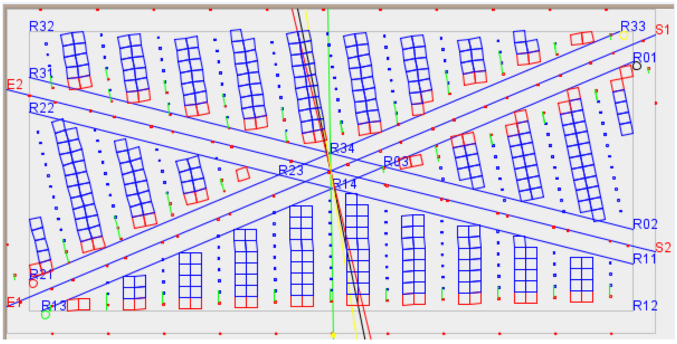
For 5 pick



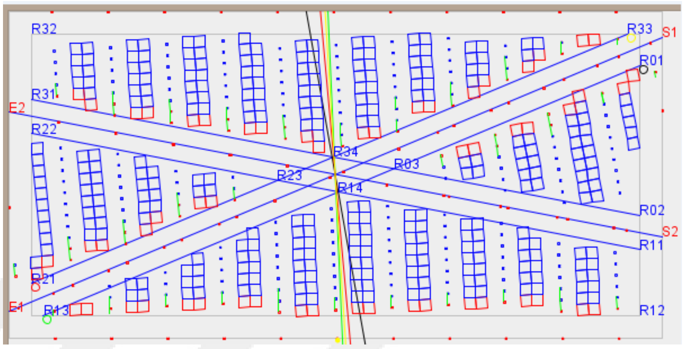
For 10 pick



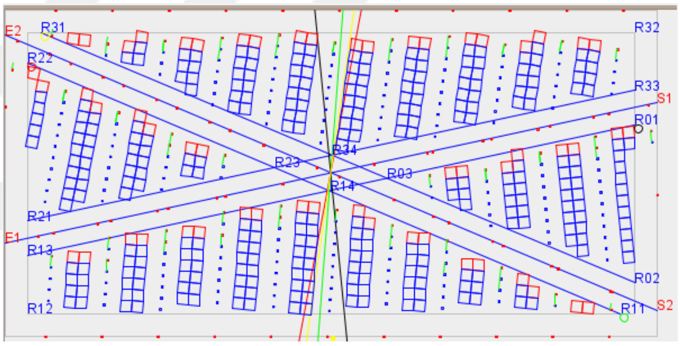
For 20 pick



For 30 pick



For 40 pick



For 50 pick

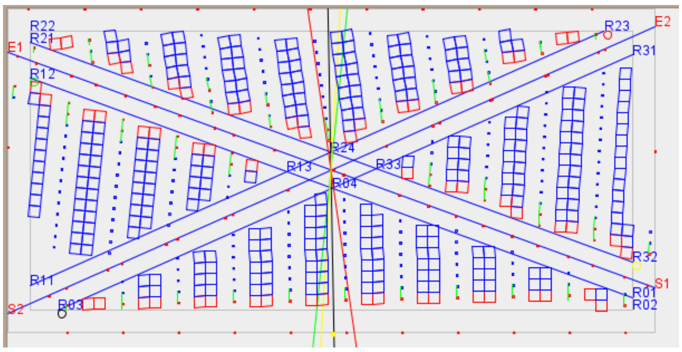
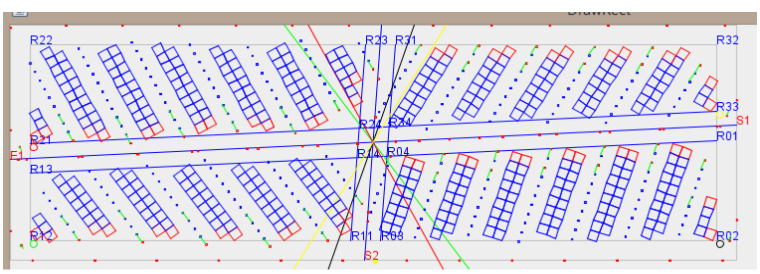


**Figure A5.2** Warehouse Representations of 2:1 Shape Ratio.

For 3 pick



For 5 pick

For 10 pick

For 20 pick

For 30 pick

For 40 pick

For 50 pick

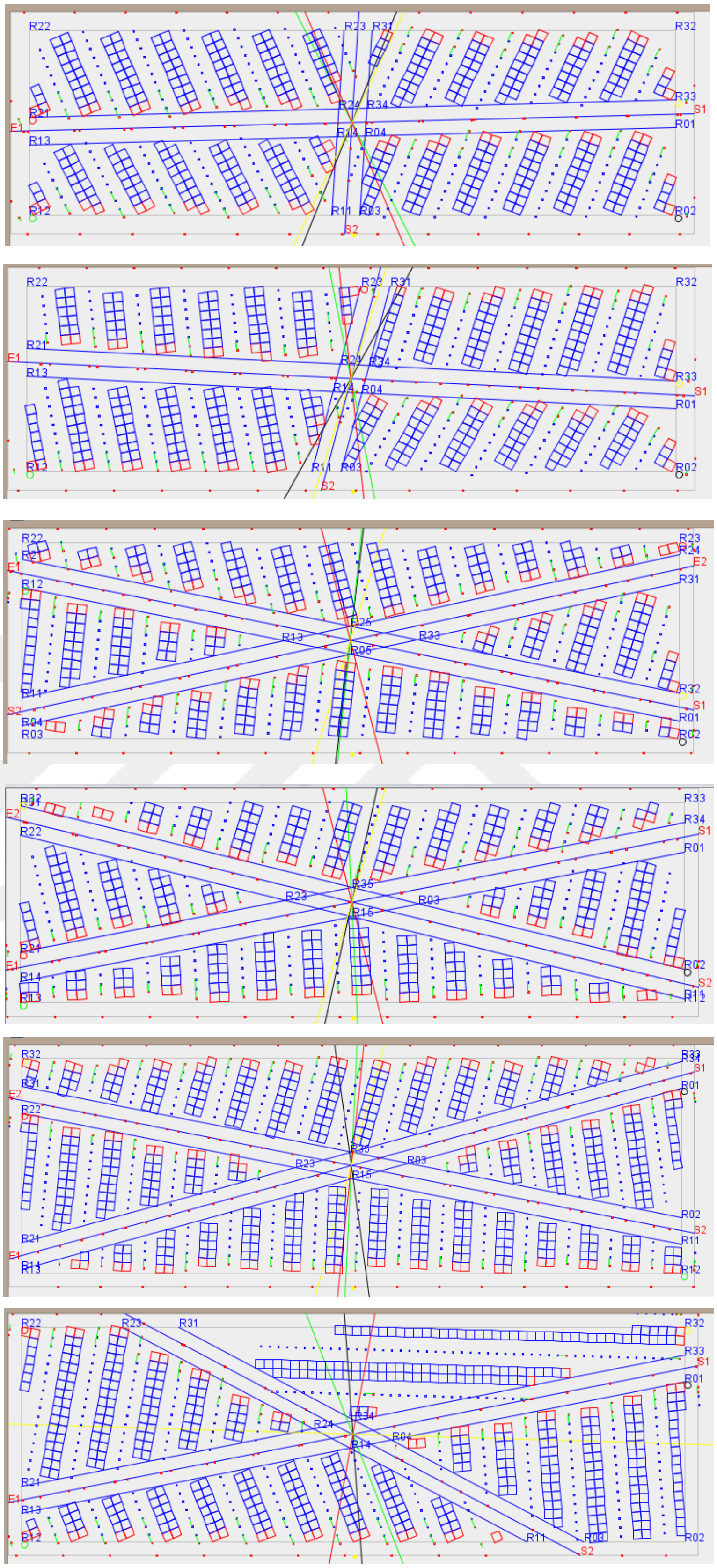


**Figure A5.3** Warehouse Representations of 3:1 Shape Ratio.

## APPENDIX 6 – RESULT TABLES

**Table A6.1** Features of Best X-shape and Design B with Equal Sizes.

| Shape ratio 1:1, Width 380 PU, Length 420 PU | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Pick List # | Capacity | m | β | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | X-shape Fitness | Design B Fitness |
| 3 | 400 | 1.29 | 46.4 | 118.8 | 118.8 | 70.3 | 95.0 | 1,003.7 | 1,064.3 |
| 5 | 386 | 1.19 | 47.6 | 75.1 | 85.3 | 96.7 | 85.3 | 1,263.9 | 1,312.3 |
| 10 | 379 | 1.78 | 136.8 | 86.0 | 80.7 | 102.3 | 68.8 | 1,786.0 | 1,800.1 |
| 20 | 376 | 1.09 | 55.4 | 84.3 | 84.2 | 84.0 | 84.4 | 2,404.4 | 2,480.5 |
| 30 | 378 | 1.46 | 48.4 | 86.9 | 96.7 | 86.3 | 108.9 | 2,878.9 | 2,882.4 |
| 40 | 377 | 1.01 | 39.3 | 70.6 | 89.9 | 69.2 | 75.5 | 3,068.3 | 3,135.7 |
| 50 | 374 | 0.00 | 140.5 | 4.8 | 143.4 | 113.2 | 83.7 | 3,316.1 | 3,299.3 |
| | | | | | | | | | |
| Shape ratio 2:1, Width 580 PU, Length 290 PU | | | | | | | | | |
| Pick List # | Capacity | m | β | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | X-shape Fitness | Design B Fitness |
| 3 | 383 | 1.56 | 86.9 | 110.8 | 65.3 | 70.2 | 128.3 | 1,012.67 | 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 | 3,55.96 |
| | | | | | | | | | |
| Shape ratio 3:1, Width 730 PU, Length 240 PU | | | | | | | | | |
| Pick List # | Capacity | m | β | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha_4$ | X-shape Fitness | Design B Fitness |
| 3 | 388 | 1.43 | 97.0 | 109.1 | 53.3 | 61.5 | 121.6 | 1,070.274 | 1,207.640 |
| 5 | 388 | 1.46 | 95.3 | 111.6 | 63.3 | 67.3 | 115.0 | 1,411.955 | 1,586.260 |
| 10 | 387 | 1.58 | 102.0 | 118.7 | 78.9 | 83.9 | 107.5 | 1,957.411 | 2,224.240 |
| 20 | 327 | 1.81 | 156.1 | 96.7 | 96.0 | 75.5 | 107.1 | 2,548.077 | 2,981.780 |
| 30 | 349 | 1.20 | 24.8 | 24.8 | 102.6 | 87.2 | 106.4 | 3.451,780 | 3,451.780 |
| 40 | 359 | 1.12 | 26.06 | 82.03 | 92.82 | 96.12 | 105.7 | 3.744,560 | 3,744.560 |
| 50 | 396 | 1.22 | 38.32 | 85.85 | 68.32 | 100.6 | 1.66 | 3.938,520 | 3,938.520 |

**Table A6.2** Features of Best X-shape and Design B with Equal Capacity.

| Shape Ratio | Width | Length | Area | Pick List Size | Capacity | Avg Tour Length | Deviat. of tour | Extra Area |
|---|---|---|---|---|---|---|---|---|
| 1:1 | 440 | 430 | 189.200 | 3 | 514 | 1,132.320 | -6.4% | 18.5% |
| | 440 | 440 | 193.600 | 5 | 514 | 1,440.564 | -9.8% | 21.3% |
| | 440 | 440 | 193.600 | 10 | 511 | 1,997.546 | -11.0% | 21.3% |
| | 450 | 430 | 193.500 | 20 | 513 | 2,758.131 | -11.2% | 21.2% |
| | 450 | 430 | 193.500 | 30 | 514 | 3,321.361 | -15.2% | 21.2% |
| | 450 | 430 | 193.500 | 40 | 509 | 3,657.479 | -16.6% | 21.2% |
| | 440 | 450 | 198.000 | 50 | 513 | 4,049.366 | -22.7% | 24.1% |
| 2:1 | 620 | 310 | 192.200 | 3 | 500 | 1,104.290 | -2.0% | 14.3% |
| | 640 | 330 | 211.200 | 5 | 520 | 1,464.271 | -7.8% | 25.6% |
| | 640 | 320 | 204.800 | 10 | 509 | 1,992.681 | -8.3% | 21.8% |
| | 650 | 330 | 214.500 | 20 | 514 | 2,808.315 | -11.4% | 27.5% |
| | 640 | 330 | 211.200 | 30 | 515 | 3,356.055 | -12.7% | 25.6% |
| | 640 | 330 | 211.200 | 40 | 507 | 3,747.147 | -13.1% | 25.6% |
| | 640 | 330 | 211.200 | 50 | 505 | 4,144.813 | -16.5% | 25.6% |
| 3:1 | 780 | 260 | 202.800 | 3 | 508 | 1,176.907 | 2.5% | 15.8% |
| | 730 | 290 | 211.700 | 5 | 513 | 1,569.349 | 1.1% | 20.8% |
| | 730 | 290 | 211.700 | 10 | 505 | 2,181.195 | 1.9% | 20.8% |
| | 840 | 270 | 226.800 | 20 | 512 | 2,975.779 | 0.2% | 29.5% |
| | 810 | 280 | 226.800 | 30 | 512 | 3,451.780 | -2,03% | 29,45% |
| | 810 | 280 | 226.800 | 40 | 517 | 3,744.560 | -5,79% | 29,45% |
| | 790 | 270 | 213.300 | 50 | 508 | 3,938.520 | -7,36% | 21,75% |