THE DESIGN OF A PACKAGING LINE IN A MANUFACTURING COMPANY.

MASTER’S THESIS

by

Łukasz Kawczyński

LITH – IPE – EX - - 05 / 762 - - SE

Warsaw, September 2005
THE DESIGN OF A PACKAGING LINE IN A MANUFACTURING COMPANY.

Lukasz Kawczynski

In today’s competitive global economy, the focus is on faster delivery of orders at lower total costs. In this paper we are interested in several aspects of order picking systems. We examine the influence of station layout, storage policy, picking policy and sorting solution on order picking system performance. On each of the analysis we consider a few solutions. We determine influence of different number of station in raw on picking system performance. We design the replenishment system supported by kanban philosophy with implemented economical order quantity (EOQ) and reorder point (ROP). The picking system is designed for assumed product’s demand values. The assessment of each of the solution is done through Arena simulation model. The results show that properly designed station with reasonable storage policy and implemented batching policy brings significant raise in order picking system productivity. In addition, we found that proper sortation system logic allows for more equal workload and reduction of maximum queue lengths. The results offer solutions to managers looking to implement improvements in order picking systems.

Order picking, mail order, station layout, packaging line, replenishment, storage policy, sortation, picking zone.
THE DESIGN OF A PACKAGING LINE IN A MANUFACTURING COMPANY.

MASTER’S THESIS

by

Łukasz Kawczyński

Supervisor at Linkoping Institute of Technology:

Ruth Sommar

LITH – IPE – EX - - 05 / 762 - - SE

Warsaw, September 2005
Executive summary

In today’s competitive global economy, the focus is on faster delivery to a customer of small more frequent orders. Order picking activity is critical process in mail order companies. The fast and effective order picking is new challenge. In this paper, we are interested in designing under assumed demand data order picking line.

In essence, it has been studied the best way to pick all orders. To do this, it is necessary to investigate order picking process. This investigation involves station layout, storage policy, picking policy, replenishment process and sorting solution. It is also necessary to also investigate size and number of the stations to be installed. In station layout analysis, five station arrangements are developed. Storage policy analysis considers products to bins assignment algorithms. Benefits of batching algorithm are depicted in picking policy analysis. In replenishment process analysis, Kanban with economical order quantity and reorder point is developed. Sorting solution is investigated in order to provide the best transportation of boxes. The aim of the analysis is to provide solution that requires the shortest conveyor length and the smallest number of picking staff.

A simulation model in Arena was developed to represent the system and its actual behavior. The model was created for each of the field of analysis. The simulation was a tool to choose the best from considered solution. Several experiments were conducted to obtain all the desired information.

The solution developed gives significant benefits. The number of picking staff was reduced by 46%. The length of the conveyor required as well dropped. The reference solution required 265 meters of the conveyor, while solution developed needed only 104 meters. The analysis developed shows clearly that proposed solutions are better in basic performance like number of orders and units served, number of staff required, length of the conveyor needed, single picker utilization and total time in the system.
Acknowledgements

I want to thank my supervisor at Linkoping University: Ruth Sommar for helpful science advises and a lot of patient during long phone talks. I want to thank as well my supervisor at Warsaw University of Technology: Krzysztof Santarek for academics support.

I want to thank my parents for their support in hard moments and creating the possibility to reach all this goals that I have never been dreaming about. I want to thank my girlfriend for her patient during this endless task. Finally, I want to thank my friend who sold me to go for Erasmus exchange.

This is my beginning…

Lukasz Kawczynski
7th October 2005
# Table of content

Executive summary......................................................................................................... 4  
Acknowledgements......................................................................................................... 5  
Table of content .............................................................................................................. 6  
1 Thesis introduction .................................................................................................. 8  
   1.1 Background ...................................................................................................... 8  
   1.1.1 Literature overview................................................................................... 8  
   1.2 Problem formulation .................................................................................... 9  
   1.3 Aim and thesis’ objectives .......................................................................... 9  
   1.4 Scope .............................................................................................................. 10  
   1.5 Limitations ..................................................................................................... 10  
   1.6 Methodology .................................................................................................. 10  
   1.7 Outline of thesis report .............................................................................. 14  
   1.8 Scope definitions .......................................................................................... 14  
2 Description of the problem.................................................................................... 20  
   2.1 Detailed problem formulation .................................................................... 20  
   2.2 Detailed packaging line requirements ....................................................... 23  
   2.3 Order picking system acceptance criteria ................................................ 24  
3 System configuration ............................................................................................. 25  
   3.1 Mechanical assembly system characteristic ............................................ 25  
   3.2 Manual assembly characteristic .................................................................. 28  
4 The reference model .............................................................................................. 32  
   4.1 Reference station layout ............................................................................. 32  
   4.2 Reference storage policy ............................................................................ 33  
   4.3 Reference picking policy ........................................................................... 34  
   4.4 Reference replenishment process ............................................................... 34  
   4.5 Reference sorting solution ......................................................................... 34  
5 The order picking system analysis ........................................................................ 36  
   5.1 Number of zones .......................................................................................... 36  
   5.2 Station layout ................................................................................................ 38  
   5.2.1 Revision of the possible station arrangements ....................................... 38  
   5.2.2 Fast moving products station layout ...................................................... 40  
   5.2.3 Number of fast moving products U shape stations............................... 52  
   5.2.4 Medium moving products station layout .............................................. 60  
   5.2.5 Number of medium moving products stations ...................................... 64  
   5.2.6 Slow moving products station layout .................................................... 70  
   5.2.7 Number of slow moving products stations............................................ 77  
   5.3 Storage policies .............................................................................................. 82  
   5.4 Picking policies ............................................................................................... 90  
   5.5 Replenishment process ............................................................................... 103  
   5.6 Sorting solution ............................................................................................. 114  
6 Conclusion........................................................................................................... 121  
7 Further research ................................................................................................... 122  
REFERENCES ........................................................................................................... 123  
Appendix 1. Monthly demand data............................................................................. 125
Appendix 2. Demand skewness ................................................................. 143
Appendix 3. The economic convex hull algorithm .................................. 144
Appendix 4. The 6-D SFC θ algorithm ....................................................... 145
Appendix 5. Fast moving products zone, U shape, 3 stations in the raw arrangement documentation (Siman language) ................................................. 147
Appendix 6. Fast moving products zone, U-shape, 4 station in raw arrangement documentation (Siman language) ............................................. 166
Appendix 7. The horizontal, single station layout with diagonal storage policy (Siman language) ................................................................. 185
Appendix 8. The replenishment model documentation (Siman language) ...... 207
Appendix 9. Final model documentation (Siman language) ....................... 215
1 Thesis introduction

1.1 Background

In many companies order picking process is becoming year by year more important. Successful order picking is necessary for fast order shipment to customer. On market which is full of competitors, fast shipment becomes order qualifier. In case of some specific products – for instance pharmacy, it might become order winner.

The order picking systems in mail order companies was chosen as a subject of a thesis, because there is none publication which gathers all the aspects of picking process. In the picking process there is still field for improvements.

1.1.1 Literature overview

Picking zone configuration has been widely described for warehousing. There is a very few literature describing different shapes, sizes and configuration of manual picking systems. Especially, analysis of size of the station seemed receive little attention. The issue of product location for the rectangular or linear storage racks has received much attention. The size analysis of the picking zone is developed by Petersen, C.G. (2002). Petersen, C.G (2002). Author gives as well analysis of picking zone configuration. The author considers as well different storage policies within the zone. The clustering algorithm for products assignment to particular zone is given by Chin-Chia Jane and Yih-Wenn Laih, (2003). The authors introduce the idea of synchronized zoning system. Che-Hung Lin and Iuan-Yuan Lu, (1999) among others examined order picking strategies. Picking strategies very widely were examined as well by Hinojosa, A. (2003).

Several authors have looked into routing policies, among others: Petersen C.G., et.al. (2004), Hwang, H, et. al. (2004). Caron, F., et. al. (2000) have discussed different server home base location in connection with routing policies.

Byuing-In, Kim; et. al. (2003) looked into replenishment process of picking zones. The order picking performance as the result of the product skewness was analyzed by the Petersen, C.G.; et. al. (2004). The authors were able to find and prove influence of business parameters on the order picking performance.

Brynzer, H., et. al. (1994) shows the methodology of “zero base analysis” implemented for the order picking process. The aim of analysis is to improve the process by eliminating unnecessary activities.

In summary, previous studies consider order picking activities as a set of separate parts: replenishment, storage, picking and sorting. None of the studies discussed these aspects simultaneously as a one set. We do this in this thesis.

1.2 Problem formulation

The problem is to design a line in the packaging department in a pharmaceutical or cosmetic company. The hypothetic company runs the business on the basis of mail order. There is a need to pack the products according to specific customer order. Each order is customized and unique. Customer does not have any limitations in placing the orders (no quantity limit, no minimum order border, nor order value). We are going to analyze number of staff needed to serve the line as well as the equipment needed. Staff is going to be divided according to the position performed. The result of the project should be the rough layout and a simulation of a line under consideration in Arena. The order picking system covered by the layout has to be able to serve demand data depicted in Appendix 1. The demand is a result of the seasonal picks and bottoms. The company might influence the demand in monthly periods – it means that each month company settles new offer of products. The offer is changing from month to month, which is depicted in Appendix 1.

1.3 Aim and thesis' objectives

The aim of the thesis is to provide project of the packaging line. The thesis objective is to provide for assumed sales values the for best packaging line layout from considered. The critical question is how to design and develop order picking line. What are the necessary requirements for a packaging line to provide for the company flexibility and effectiveness of acting? The present work shows step by step how to deal with such a problem, from early formulation to the final layout specification. The packaging line problem is going to be divided into smaller problems. There will be drawn relationships between different issues and areas. The objective of the thesis is to provide the layout of the line under consideration, which will be able to handle assumed demand. In order to provide better understanding of the picking process there is developed an Arena simulation model, which is as well a base to evaluate considered solutions.

We are interested here in the concurrent issues of (1) picking zone configuration and performance, (2) product location – storage policies, (3) picking strategies, (4) convey system configuration, (5) replenishment system.
1.4 Scope

The thesis subject requires knowledge from various fields. There will be shown the layout of the line including staff so the thesis will contain elements of work organization. The report provides single worker post space layout, so the thesis contains the elements of work place organization. The work place organization issues have to be supported by ergonomic and anthropology data. Finally, the report includes the general layout of the line, so the knowledge required here is from the area of manufacturing/assembly operations. There is a need to interpret an input and output data, so there are required data analysis skills. Analysis concerns appropriate process scheduling, to achieve this there is needed knowledge about process scheduling. Finally, solving the problem requires the knowledge of statistics and mathematical analysis. In order to create the model and better visualize the results there are required skills in Arena software.

1.5 Limitations

The thesis is done on some level of details. It is not author’s intention to provide very detailed layout, including specific technical issues. One of the most important characteristics of such project for the company is total cost. The cost part of the project is very difficult and requires high level of details. Being aware of that, the cost is not a part of our analysis. We consider the cost only quantitative in meaning of staff and length of the conveyor. We choose solutions by assessing the number of staff needed and length of the conveyor required. By implementing above logic we receive final solution, which is the best solution from considered. The staff that is analyzed here is the shop floor workers. The report does not cover white collar workers.

Only basic health and safety requirements are considered. Each country specifies on its own detailed requirements in this fields. These requirements are covered in health and safety at work legislations. Similar situation is with the law requirements. Only general basis are covered by this report.

1.6 Methodology

The thesis aim is to show the design of mail order packaging lines. The thesis analyzes the critical processes run during picking orders. At the beginning it is necessary to get and define input data and limitations. The input data – the table with monthly product demands, is created by statistical variation, and is covered in Appendix 1. The other data as number of orders are being assumed further. The product demand data were created by sequential statistical variation. In order to maintain some differences between quantities of high and low demand product we grouped the products into groups. Respecting the share of each group in the total demand, the values of demand were created among the group. The number of products between groups was varying.
This methodology allowed us to have a few high demand products and many low demand products.

We start the thesis with a literature overview. Then we come up with scope definitions, which give reader basic knowledge and ability to understand the problem. Defined key words are used along whole thesis, so it is very important to understand this part. We continue with a detailed description of the problem. We add more details, and show the entire problem. This is an integral part together with the project requirements. We set market and customer requirements, which are the internal part of the criteria of acceptance. The phase allows us to specify all deliverables to be researched and reported.

Since the picking line is a sophisticated system, we divide the analysis into smaller pieces. We lead bottom-up analysis. On each of the steps, except replenishment section, we create the model for each of the solutions considered, we examined its performance and then we are choosing the best one. For the replenishment section we show how to implement kanban system. We start our analysis from the station layout, which is the smallest part of the system. We find out in simulation an influence of station shape to the picking performance. In the second part of the station layout section, we analyze influence on performance of multiple numbers of stations in the line. Then we follow with the storage policy. We specify the results of different storage policies implementation. In Section 5.4 we consider different picking policies. Then we continue with replenishment process analysis. In order to complete whole line we analyze the transportation / sortation solution. The analysis on each of the steps is done separately for each of a product zones. The number of products zones is specified in Section 5.1.

The simulation models that we develop are descriptive models. The models developed below are using probability functions in order to describe simulated situations. Each model simplifies in some way the real process. In each model the list of basic processes is simulated. Each process is using statistical data gathered from the literature. The list of the assignment and decision modules is implemented in order to characterize system logic. The simplifications appear in lasting times of processes and in the architecture of system logic. The details about the models architecture is given in each of the sections, where the simulation was used.

Moreover in order to have a basis for the comparison of performance, we create a reference model. The reference model is created in each of the five fields of analysis – station layout, storage policy, picking policy, replenishment process and sorting solution. On each part of the analysis we refer to the reference model. The purpose of the reference model is to create the base for the comparison. In the analysis the reference model is used for comparison of performance. On each of the stage of analysis except the replenishment section, we develop few solutions. Performance of each of solutions is compared to the reference model performance. The solutions are as well compared between each other. We discuss advantages and draw backs of each of the solutions.
The referenced model is not an existing model. The reference model is created without any deeper analysis of demand pattern neither considering demand skewness. Hence, there is almost 100% probability that reference model will not be the best solution and there will be a lot of to improve. The performance of the reference model is at the beginning of the analysis unknown. We gather reference model performance, as we build in Arena simulation model. We relate our solutions to the reference one for the station layout, the picking policies, the storage policies, replenishment process and the sorting solution. The analysis proceeds from the bottom to the top, which means that we start with the analysis of the lowest level of the system, which is the station layout. Further we analyze the picking policies, then storage policies, replenishment process and we end the analysis with the sorting solution. At the end of each section under consideration we chose one best from considered solution. The bottom-up methodology is depicted in figure 1.1. The best solution from the previous section is the input data to the next section. For instance we chose through the simulation the best station layout; then we use the best station layout in next section in order to analyze storage policy.
The main criterion that we evaluate is the amount of staff needed. Each of the solution means particular number of blue collar workers. The lower number of workers is required the better solution is, because it means that workers are working more effectively. The second aspect that we evaluate is average and maximum number of orders in queues. This parameter determines the buffer sizes between stations. The buffer sizes between stations are realized by rising up the space on the conveyor. The lower average and maximum number of orders in the queue, the better solution is, since less conveyor space is required. Hence, we assess as well the utilization of the pickers. We aim at maximum picker’s utilization. If the single picker utilization is too
low it means that picker still has too much idle time, and still there is a place to improve the process.

1.7 Outline of thesis report

We start thesis with scope definitions, in order to build up subject understanding. We use keywords from that part in whole paper. Then in Chapter 2, we follow with detailed problem formulation. In Chapter 3, we summarize possible system configuration – manual and automatic one. The knowledge from this chapter is used along further chapters. In Chapter 4, we continue with reference model formulation. We determine reference model in each of the five fields of analysis: station layout, storage policy, picking policy, replenishment process and sorting solution. Reference model is used in each of the following sections for comparison. In Chapter 5 we follow with detailed system analysis. We start in Subsection 5.2.1 from determining number of zones and units split of each zone. This section determines one of the most important input parameters, which influences whole further analysis. In Subsection 5.2.2 we start with station layout analysis for fast moving products zone. We describe possible solutions that are examined. We compare the performance of each of the solutions to the referenced one determined in Chapter 4. The best solution from considered becomes input to Subsection 5.2.3 in which we analyze number of stations in fast moving products zone arranged in line. We continue with similar analysis of station layout and number of station in the raw for medium and slow moving products in Subsections 5.2.4 – 5.2.7. In Section 5.3 the best solutions from Section 5.2 is used in order to analyze storage policies. Section 5.4 analyzes the picking process for the system chosen in previous sections. Then we proceed in Section 5.5 with the replenishment process analysis, which develops replenishment strategy. In Section 5.6., we analyze sorting device for solutions chosen in previous sections. Chapter 6 and 7 ends the paper with conclusion and further research.

1.8 Scope definitions

SKU – storage keeping units.

Size of line SOL – number of products in company range. It is a variety of products that company offers to its customers. This quantity might differ from month to month. The amount of products depends on the strategy of the company. It is said that 20 % percents of the products makes 80 % of the business. The rest of the line (80% of the products) is only “decoration”. The company might influence on which products are right now selling through the marketing activities. For the thesis I assume the monthly size of line on the level of 1400 products. Size of line is expressed and measured monthly.
**Bin** – the place on the station where the product is stored, just before packing into box. Bin is integral part of a shelf of flow rack. Bins inside the segment have the same dimension. The number of bins is varying depending on the segment. In the bin products are usually kept in packers.

**Station** – the organized arrangement of bins. It is also separated area in a line. One box is gradually packed by stopping on appropriate stations. The number of the bins on the stations is going to be established in further analysis and is going to vary depending on the segment.

**Segment (zone)** – organized, gathered in group stations. A segment has stations, which contains products with similar daily sale. The number of segments might vary and strongly depends on size of line and quantity of bins per station.

**Picking list** – list of product in an order.

**Sorter** – a set of conveyors which transport the boxes into segments. The sorter is responsible for delivering each box into required segment (and station). When the box has visited all needed stations (the order has been satisfied) the sorter transports the box to the exit of the system. The sorter from technical point of view is the set of rollers, diverters, stoppers, laser scanners, rubber conveyor belts, photocells, scales, pop-ups and label printers.

**Bin filler** – the person whom main task is to transport a packer with products into bins. Bin filler moves the packers from the euro pallet place into appropriate bin on the station. The euro pallets are located in the warehouse. In order to transport cartoons bin filler uses trolley.

**Picking staff** – the person that is working on the station. The main task is to put the product from the bin into the box on the basis of the order.

**Replenishment** – department responsible for continuous supplying bins with the products. The main workers are bin fillers.

**Flow rack** – lean set of the racks. From one side flow rack is refilled by bin filler, and on the other side the picking staff put the products into the boxes. Each shelf is leaned in order to the cartoon has a possibility to slide down into the bin. The lean is approximately 10-15 degree. The angle can not be to big, cause the cartoon might fall out of the bin. The flow rack is deep enough to contain up to four cartons. The cardboard has the acceleration due to gravity. The idea of the flow rack is showed in figure 1.2.
Static shelf – a set of the racks, which are parallel to the floor. The static shelf is dedicated for the bins with low daily sale products. The static shelve might contain only single carton. The idea of the static shelf is depicted in figure 1.3.

Productivity – we can consider three dimensions of productivity. We can look at the productivity at very general level and it expresses mainly the performance of the sorter as a machine. The unit will be here the number of boxes per hour. Second point of view is the productivity of the segment, which is equal set of the lines productivity. The dimension here is also the boxes per hour. The lowest level of looking at the productivity is the picking staff level. Each person assembling the order has average productivity. The dimension here is number of picks per hour, per person. It means that this person can pack specific number of units per hour.

Pop-up – cross roads of conveyor which has up to three junctions. The box when goes through pop-up does not slow down. The pop-up guaranties high performance of the system. The pop-up from technical point of view is the set of the spinning small rings. The rings are grouped in rows. The rings are changing the angle relative to the axis going through the middle of the ring, which is at the same time perpendicular to the surface of conveyor. The rings are starting to change the angle just when the box is
passing through pop-up. Each row of the rings is changing the angle in different way. Each row has various delays and angle in relation to the box passing through. The pop-up can not change the movement of the box for 90 degrees.

**Diverter** – device which changes direction of box movement for 90 degrees. It is situated between regular rollers of the conveyor. It raises the box and the rollers mechanism transports it to next conveyor. After transporting the box the rollers are lowered in order to make for the next boxes possible to go straight. The main idea of the diverter is showed in figure 1.4 and in figure 1.5.

![Diverter view from top](image1)

*Figure 1.4. Diverter view from top*

![Diverter view from side](image2)

*Figure 1.5. Diverter view from side*

**Stopper** – the device which does not allow boxes to lump together. The stopper does prevent boxes to approach to each other too close. It is especially necessary in case of areas where there might boxes traffic jams appears. The stopper is located between rollers of the conveyor. In order to avoid lump of boxes the stopper block the boxes one by one. In order to make it possible to work efficiently the stopper is connected to the set of photo cells. The idea of how stopper is working is showed in Figure 1.6 and in Figure 1.7.
Mechanical assembly system MAS – it is a device which makes possible semi automatic packaging. It is just semi automatic, because machine still requires manual refilling channels with the products. MAS is constructed from the main conveyor and the set of the channels with the products. Products are through away from the channels by the electric ejectors. Afterward they are placed on the main conveyor, which is passing all of the channels. The channels are formed in the shape of A capita letter. At the end of the conveyor is a dump station, where the products from the conveyor are dropped into the box. The device creates virtually the gap on the conveyor and assigns it to particular order. There is option to divide this gap into two gaps. It will allow separating heavy product from fertile. It is quite useful cause at the end of conveyor the products are dropped into box from around 30-45 centimeters. In the situation when products are not separated, heavy products might squash fertile products. If we consider the division of the gap, the heavier products are dropped to the box as a first; the probability that fertile products will be damaged is much lower. The most important objective of the auto picking machine is great reduction in personnel costs and shorter transit times of orders. One of the most important advantages is extremely high accuracy. The percent of the mistakes is close to zero. It means that a machine almost does not make any errors. The replenishment of the machine is as well very easy. The process of refilling the MAS is been designed in order to be as much efficient and as fast as it is possible. Moreover refilling activities might be done without disturbing the work of the machine. More detailed description and analysis is going to be done below.

Dumb station – it is the part of the mechanical assembly line where the products are dropped into the box. The products are through away from the channels to the main MAS conveyor. After whole order has been complete the products are dropped into the box. The difference between the high of the conveyor and the surface of the bottom of the box is around 45 centimeters. The products are dropped from this distance from the
edge of the conveyor to the box. The box is transported within the main sorter conveyor. Optionally there is possibility to install just before dump station acceleration tunnel. It accelerates the box just before refilling it. This mechanism makes possible to achieve bigger number of the boxes served per hour. The boxes are gathered in the queue before the dump station. Then they are accelerated separately, and just before dropping down the order they are stopped for a second. After whole order is been dropped into the box the box is once again accelerated. The tunnel mechanism allows adjusting the performance of the MAS.

*Warehouse control system* – it is a system, which allows controlling the issues connected with the stocks management in the company. The system contains the software and hardware. By the software we understand here the program with interface. The hardware is the devices as for instance radio terminals.
2 Description of the problem

2.1 Detailed problem formulation

The thesis objective is to design the packaging line for an imaginative mail-order company. Company divided the sale year into twelve periods. In each period the offer of the products is changing. There is changed price list of the products and as well the exposition of it in the leaf let. As a result monthly sale is varying. Other factor, which influences the demands, is seasonality. There are some products which are sold only during the summer, and the winter sale is almost zero. Created monthly sale sample of the products might be found in the Appendix 1. The size of lines is established to be 1400 products per year.

From the rough analysis of the data from Appendix 1 we might notice the distribution of the sale along the year. It is visible that there are two picks. The main one is in December. The bottom of the sale on the figure 2.1 is in February and March.

![Number of units sold](image)

*Figure 2.1. Monthly sale.*

Another important issue is distribution of the sale within the month. In order to have a rough view we construct the product line histogram. The samples of the histograms are depicted in figure 2.2, figure 2.3, figure 2.4. The full list of histograms might be found in Appendix 2.
Figure 2.2. Demand skewness in fourth quarter.

Figure 2.3. Demand skewness in third quarter.
It is visible from figures 2.2, 2.3, 2.4 that around 50% of the line are the products with lower demands. The conclusion from above figures is that the company has 50% products in the offer which are not sold in large quantities. Looking at the Appendix 1, we can see that part of the line is not in the offer since monthly sale is equal to zero. In December there are around 480 such products. If summarized above, around 1100 products in December are sold in small quantities or are not sold at all. The rest of the products are sold within higher quantities. From above histograms we can observe that there are only a few top sale demand products. It means that for the company really profitable is only couple of products, sold in large quantities. For the packaging line it means division of the products into groups and as a result segmentation of the products. The approach is going to be analyzed further. Average month in our analysis has twenty working days.

The important input parameter is number of orders placed. This value gives us the number of units per order. We assume that the average order size is 6 products. The order size is characterized by poisson distribution with the mean value of 6. The number of products per order leads the analysis to the sizes of the boxes. In order to establish size or sizes of boxes the company should analyze the dimensions and volumes of products. Knowing the number of the orders, having the samples of the orders and specifications of the products packaging the company might specify required sizes of the boxes and the number of box size versions. In our analysis we will simplify the reality and we will not consider the number of units being packed of particular product. This parameter is important for analyzing size of the box required. This parameter from order picking system point of view that we analyze has small influence, since for us it does no matter if picker within the same tour picks one piece
of product A or two pieces. This issue might become crucial if we would get close to the pickers’ capacity. We assume that the number of units of each product being ordered is far away from picker’s capacity, so therefore is negligible. The sample of the orders in connection with products characteristics specifies the possible sizes of the boxes. Company has to analyze the volume of the products per order, and some buffer for the spaces between the products in the box. The result of the analysis should be the list of possible box sizes. The company should consider five, six box sizes.

For the purpose of the thesis we assume number of orders. We assume as well that each order is packed only in one box. The number of orders is varying among the months and is given in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numb of orders</td>
<td>141004</td>
<td>127151</td>
<td>122936</td>
<td>160414</td>
<td>172866</td>
<td>158464</td>
<td>171539</td>
<td>148861</td>
<td>134564</td>
<td>180208</td>
<td>173397</td>
<td>185388</td>
</tr>
</tbody>
</table>

*Table 1.* The number of the orders per month

2.2 Detailed packaging line requirements

The objective is to provide the packaging line which is able to pack the orders in short delivery day system. Second important customer requirement is the quality of packed product. The products can not be damaged during transportation. The order has to be packed in the way that does not damage the products. This issue is connected with quality.

Other aspect of quality is order control. The orders have to be packed, and checked in order to ensure the quality for the customer. The control of the orders is necessity, in order to reach demanded high quality.

Project objective is also to determine the area required for the system. The packaging line can not be extremely large. The area is treated here as a parameter, but objective is to keep it as small as possible. In reality each square meter means additional square meter of land and building. The smaller area is required for the system the better it is, especially if we consider the prices of land and costs of setting up the buildings.

One of the most important objectives in reality is the economical justification. The project supposes to consider as one of the most important characteristic the costs of the packaging line. The parameter which is suitable here is the costs per unit packed. In reality this aspect are treated very carefully and are preceded with deep analysis. In the thesis, due to wide range of analysis we do not perform economical justification. We asses this parameter only indirectly – by evaluating in the options number of staff and length of the conveyor required. The objective is to keep both parameters low. In case of practical analysis the lower costs per unit packed the better system is.
Another important issue of packaging line is the flexibility of the packaging system. This is immeasurable characteristic, but it might be invaluable. Flexibility in the meaning of packaging line represents the ability to easily adapt to changed conditions. The system should give the owner the ability to easily change the system in the way that it allows to pack under new conditions. It might mean change in product range or in product sale distribution. The flexible system should make possible to user the adoption to new conditions, or make it with as small effort as possible.

2.3 Order picking system acceptance criteria

The main criteria of acceptance is the ability of the system to serve within 2 shifts – 15 effectively working hours, the amount of orders from the pick month. The other criteria of acceptance include the quality of packing. It means that products during picking process can not be damaged. Moreover, one of the most important criteria and as well very hard to precisely define is the working environment for the picker. The working place should be safe and comfortable, and the work should not make possibility for injuries and hurts. The basic requirements for the work place we take from “Human dimension atlas”, (2001).
3 System configuration

The picking system may consist of mechanical and manual assembly. In order to provide the movement of the boxes between parts of the system, there are needed devices for the transportation, i.e. conveyor belt.

3.1 Mechanical assembly system characteristic

The system cannot be fully automatic, due to the machine limitations. Some of the products do not fit in meaning of shape to the mechanical assembly machine. Mechanical assembly system (MAS) is able to serve only the products which are close to cubic characteristic or cylindrical shaped. A sample of the products is shown in figure 3.1. Mechanical assembly process is restricted by the ejector design. The ejector requires at least two parallel surfaces. If this conditions is not fulfilled the ejector is not able to throw away the product. The draw of ejector might be found in figure 3.2.

![Figure 3.1. Samples of the products supported by MAS](image)
Stage 1
inoperative,
sensor for empty channel indication
activated,
ejection sensor idle (inactive)

Stage 2
cleat moves towards product,
product is ejected,
pulses are counted

Stage 3
ejection sensor activated
(long = product, short = cleat),
pulses are counted

Stage 4
next cleat passes light barrier,
sensor for empty channel indication
activated,
ejection sensor idle again (inactive)

Figure 3.2. Mechanical assembly ejector.
Second parameter in case of mechanical assembly lines is the weight of the products. The total weight of the products in the channel of MAS can not extend 6 kilograms. This issue is connected with the quantity of the products in the channel. The smaller product is the more pieces might be put into the channel. The variable characteristic here is the dimensions within which the product is situated into channel. The number of the products in the channel is critical issue from the replenishment point of view. The more products in the channel the more effectively the channel might be refilled, which is equivalent to the amount of work required. A high number of the product in the channel means efficient refilling process, which means that fewer workers are required for a service of the machine. This is influenced by the way of placing the product in the channel. We are able to put maximal number of products into the
channel when we put products within the smallest dimension vertically. The number of
the products in the channel will be maximal under these conditions.

![Figure 3.4. Product orientation in the channel.](image)

Third parameter of the mechanical assembly lines is the dimension of the products.
MAS is able to serve the products which have height between 12 and 90 mm, width
between 12 and 200 mm, length between 40 and 310 mm.

Moreover items have to be stackable, in order to be served by the MAS. Items are
thrown away by the ejector from the height around 45 centimeters. The packaging has
to be able to survive the fall. They can not get crumple or break. Especially, glass
packaging should be tested before put into the machine. Products in the channel can
not get mutually tangled or stick to ejector.

3.2 Manual assembly characteristic

The manual system consists of the picker and the station. The task of the picker is to
pick the ordered item from the bin and place it into the box. The working area should
be user-friendly – all the bins with the products should be easily reachable.

Only one picker works within one station, so the size of the station as well as
frequency of product picking in the station does matter. In case of reference model
described in Chapter 4, there is a problem of single line balance. In the reference
solution more than one station is put in the row. It means that when one station –
picker, is faster than the other one there will disturbances appear. Under such a
condition always all the pickers working in the line will have to work with the speed of
the slowest picker. In order to tackle the problem the situation will require match
between products stored within the station and the individual speed of each picker.
Reaching the balance point is very difficult and even small change within the line might cause line imbalance, which as a result will bring lowering the productivity. The detailed analysis line configuration will follow below.

The information about the item to pick, its location and its quantity might be passed through the picking list, pick to voice system, pick to light system or pick to screen system.

**Picking list system**

The information about the ordered goods might be passed through the picking list, which is a sheet of paper with appropriate data. On the paper there should be at least the product code, the product location – including station and zone number, and the ordered quantity. The picking list is the most simplified way of transferring information in mail order companies. It is as well the cheapest solution, and does require minimal investment. The only requirement is the capable printer and some software which will connect the order system with product location.

The process of picking within the picking list might look as follows:

1. The picking list is taken out of the box by the picker,
2. The picking list is read by the picker (it might be done couple of times if there are more than one good ordered within the station),
3. The picker travels to the storage location (it might be done couple of times if there are more than one good ordered within the station),
4. The picker picks the product from the storage location (it might be done couple of times if there are more than one good ordered within the station),
5. The picker carries the product(s) back to the start point,
6. The picker drops the product(s) to the box,
7. The picker marks on the picking list picked products,
8. The picking list is placed back into the box,
9. The picker waits until next box arrival,

According to *Hinojosa, A. (2003)* picking process within the picking list is more time taking process than the pick to light, which influenced strongly the productivity. On the other hand the picking list system advantage is that it does not require online system support. The picking list sheets might be printed in advance and the system brake down will not influence the picking process. Moreover, the picking list continuously supports the picker with the information about next picking activity. The picker might take the picking list with him or her and carried it while picking. But on the other hand manual administration work usually required in case of picking list system is time consuming. *Hinojosa, A. (2003)* claims that the order picking with the sheet of the paper is a very inefficient operation with very low productivity, yielding a high incidence of errors. Hence, the picking with the picking sheet, due to low productivity might be seen as a labor extensive strategy.
Pick to voice system

Pick to voice system is another solution for order picking. The system includes headphone, remote transmitter and interface to translate data from sale system to the voice. The most expensive part of the system is the software which will translate data into voice. There is a need as well to install the radio network inside facility. But if we consider that the expense is taken once for the quite long time period, the monthly depreciation should not be so high. Moreover such a system allows keeping pickers both hands idle. It influences the productivity of the picker. Two hands idle allows picker to pick the heavy goods. “Talked voices better picks” (2004) describes case study in which pick to voice system improved picking accuracy and raised by 20% the productivity of the labor. The pick to voice system brings benefits especially when the warehouse is extensive and the pickers’ carrier capacity is high. Under these conditions the picker has to travel a lot within the warehouse and comes back to the home base point mostly when the carrier storage capacity is fulfilled. The system in case of continuous picking requires some kind of speed adjustment for the individual picker capacity. An alternative solution in case of continuous picking might be confirming button after each pick, but this solution will lower the productivity. The pick to voice system is more effective in term of productivity and quality assurance comparing to the picking list system.

Pick to screen system

The system consists of a screen placed on the packing station and some software. The software is the interface between sale system and the set of screens placed on the stations. Under the screen there are two buttons placed. One is responsible for skipping to the next pick and the second button rewinds to the previous pick. The skipping button simultaneously skips the picker to the next order. When all the products from the present order has been picked, picker pressed the skipping button, the system automatically transports to the home base point next box and displays next pick on the screen. The pick to screen system allows canceling waiting time for next order. While the information is displayed on the screen the picker might start to pick, and the sorting system simultaneously transports the box with next order to the home base point. When the picker is ready to drop the picking product, the box is already at the home base point. The pick to screen main drawback is the settled localization of the screen. The picker can not bring the screen with him or her. In case of wide picking area picker might during travel to the storage place forget the information about the item to be picked or quantity. This will cause the errors. Hence comparing to the picking list system, the picker is not able to read the information about next pick while returning after pick to the home base point. The possible solution, which has appeared already on the market, is the mobile pick to screen systems. In such a system the picker has the screen on the bangle on the hand. It allows reading the information while picking activity. Other alternative is placing couple of screens within one station. Still the pick to screen system in some cases is more effective than system with the picking list.
Pick to light system

Pick to light system is another solution for order picking systems. Mostly its popularity is concerned with its improvements in productivity and quality. Especially quality aspect is very significant. The impact of the pick to light system on the quality is considerable. The pick to light is the computer driven, paperless distributions centers system. The kernel of the system is the set of the LED readouts and the central computer. The LED readouts are used to tell the picker what to pick and in which quantity. Under each storage location there is the set of the LED readouts, which are turn on when the product from this location is needed to be picked. The number of LED readouts turned on is saying the picker the quantity of the item to be picked. After the picker picked the items, he or she has to confirm the activity by pressing the confirmation button placed below the set of the LED readouts. Feare, T. (2003) reports that pick to light system is improving the productivity by 40-50%. Moreover author claims that the picking accuracy with pick to light system is 99.99%. There are almost no errors. Hence, the training with the pick to light is extremely easy. The picker need a very few time to learn how to pick with pick to light. Pick to light system allows even to hire the illiterates. This is especially meaningful problem when in order to lower the cost company hires low educated people or immigrants, who can not read. Within pick to light system the worker need to have only the ability to count. Feare, T. (2003) says that pick to light system is especially efficient when picking high volumes of the products, so it should be dedicated for fast moving products. One of the most important benefits from pick to light system is that the picker does not have to make any decision or do any searching. The pick to light system might be coupled with the replenishment and warehouse management system. Together the three systems do all of the deciding. Pick to light system supplies the user continuously with a lot of useful data, and with connection to some analysis software it might become powerful tool. LED system might be added to all of the storage types shelving, among others: static rack, flow rack, horizontal and vertical carousel and vertical lift modules. Of course the pick to light system requires high investment. The cost of pick to light technology according to Hinojosa, A. (2003) is in the range of 125 to 175 US dollars per SKU location.

We focus our analysis on the manual system configuration. The mechanical assembly is sophisticated machine in which the user cannot adjust much. The user can only adjust breaks between the orders on the main conveyor and the order of the products between the channels. The machine performance is just the result of the engineering, within which the user can not do anything. Under manual assembly, user can directly influence performance of the system, and easily change configurations. We analyze only the performance of the manual assembly system.
4 The reference model

The reference model is divided into reference station layout, reference storage policy, reference picking policy, reference replenishment process and reference sorting solution. This division will allow us to analysis each important part of the order picking system separately.

4.1 Reference station layout

The differences of the demand volumes among product list are visible without any analysis. So it is reasonable to implement different stations configurations for different product demands. As the reference we will consider two different station types: one type for fast moving products and the second one for the others – slow moving products. The fast moving products is a group of top sale products. The slow moving products are the group of low sale products. The demand border between fast and slow moving products zone is defined in Section 5.1. The station shape and size from the logical point of view will depend on frequency of picking by the server the products from the bins. If the products are higher frequency, the station should be smaller in order to minimize traveling time, and the bins should be easily reachable. In the reference model there are two floors of the racks. On the base of Bartholdi III, J.J.; et. al. (2001) the reference station layout for the fast moving products is showed in figure 4.1.

![Figure 4.1. Reference station layout for fast moving products.](image)

In case of slow moving products the traveling times might be higher since the frequency is lower. In case of fast moving products we could not implement station layout showed in figure 4.2, because due to traveling times the picker would serve low number of orders – low efficiency of the picker. We would have to implement couple of stations like in figure 4.2 or couple of pickers within one station, in order to pack the products. In case of implementing station layout from figure 4.1 for the slow moving products probably the picker would not reach the productivity, since there would be too few products with very low frequency. The solution could be to enlarge the station layout from figure 4.1 to the size which will allow containing all the slow moving products. We should take a look on the stations layout division in the way as follows: the picker can perform some number of work; he / she can perform it by picking many times small number of products situated in small area, or he /she can perform it by picking with rarer frequency big amount of products. We should take a look on this issue as on the equity. Hence, for the slow moving products station the picking position is not needed to be as optimal as in case of fast moving products. In
the reference model there are four floors on each static rack. On the base of Hwang, H., et. al. (2004) the reference station model for the slow moving products is given in figure 4.2.

![Reference station layout for the slow moving products.](image)

*Figure 4.2. Reference station layout for the slow moving products.*

The amount of the products as well as between station configurations is given in Section 5.1.

### 4.2 Reference storage policy

The reference policy places within the aisle the products with higher frequency closer to the server home base, which seems to be logical. The policy provides minimization of the traveling routes. The reference policy is called within the aisle storage policy. On the base of Hwang, H.; et. al. (2004) the within aisle storage policy is depicted in figure 4.3. Number 1 represents area where are stored products with highest demand. Number 3 represents the lowest demand products.

![Reference storage policy – within aisle.](image)

*Figure 4.3. Reference storage policy – within aisle.*
4.3 Reference picking policy

The reference picking policy is strict order picking within picker to part system. It means that in each moment of time there is only one order served, one item picked, and the picker travels within the station to the product location in order to pick desired item. The picking information is passed by the list of the picks on the paper sheet, which is placed in each box. The picking list contains the product symbol, the product name, the product storage place – the bin symbol and the ordered quantity. In order to pick the item, the picker has to take out of the box picking list, read the information, travel to the place of storage, pick the item, travel back and put it in the box. In order to start packing next order picker has to push the box to the moving part of the conveyor.

4.4 Reference replenishment process

The reference replenishment process is done by manually driven carts. Staff refills the bins by carrying the boxes with the products from the warehouse, where the products are stored on the pallet locations. There is one kind of product carrying at time by the refilling staff. There is possibility to carrying number of boxes with the product at time.

4.5 Reference sorting solution

Since we do not know the exact performance of the picking station the figure 4.4 does not specify the number of lines, nor the size of the station – the number of bins. This is realized by the dots in figure 4.4. The lack of the knowledge on the exact stations performance did not allow us to specify exact number of lines. The number of station in the line is assumed to be three for fast moving products, and one for slow moving products. Stations within the line differ from each other. It means that they have different, unique sets of products. The content of the station between lines is exactly the same. The reference model has exactly one entrance and exactly one exit. Each order is forced to pass fast moving zone. In case of slow moving zone there is a by-pass. The by-pass solution allows orders skipping the slow moving zone and passing directly to the end point of the line. It might be explained as a solution, which reduces number of orders passing slow moving zone. It seems to be especially justified if we take into consideration the frequency of the products. There is much higher probability that each box contains the products from the fast moving zone that from the slow moving. That is why boxes are forced to pass the fast moving zone. The probability is still unknown exactly because we do not know the size of each station type exactly. It is analyzed in Subsections 5.2.3, 5.2.5 and 5.2.7. At the each line of slow moving products there is only one station. Cause of scanners which are placed on the conveyor and bar codes on the cartons, boxes are entering the slow moving zone only if they have to; otherwise they pass to end point directly. The system is evidently push system kind, in case of temporary lack of the capacity in stations the whole system will stop.
The reference sorting solution is given in figure 4.4. The reference sorting solution was proposed by Chin-Chia, J. and Yih-Wenn, L (2003), since the division of the picking system into zones bring significant improvement in whole system productivity.

Figure 4.4. Reference sorting solution.
5 The order picking system analysis

5.1 Number of zones

In the literature there is not much said about the optimal number of zones. There is no algorithm, no pattern, even no particular methodology in solving this problem. Petersen, C.G.; et. al. (2004), show the percents of savings done over random storage by number of zones in relation to the pick list size. As the result they find out, that the number of zones provides the meaningful savings mostly in case of small pick list sizes – around 5 products. The largest gap in savings over random storage was observed between two and three zones. This difference was around 3-4 % of savings in case of two and three storage classes. The difference between three and four storage classes was much lower – around 1-2 % of savings over random storage. In case of small picking lists – 5 products, the savings over random storage hesitate from 22 up to 26 %, in case of two and four storage classes respectively. The longer pick list – 30 SKU, resulted in 11-14 % of savings respectively. The percents of savings were related to random storage policy, within one zone. Authors examined only situation when we have two, three and four storage classes. Most of the authors just assumed the number of zones without any deeper analysis.

The number of zones is very important factor because it does influenced the zone pick list size, which directly influenced the performance of the system. The shorter pick list size the shorter picking time within the zone or station. Moreover, higher number of zones in connection with flexible sorting solution might lead to the situation where most of the boxes will visit for instance three from four zones. In case of presented solution in referenced model the higher number of zones will probably lead to better usage of station, since the stations will become smaller, and the boxes will visit those stations which are really necessary, not all of them in the raw. Hence, the number of zones depends strongly on the pick list size. The number of zone should be different for the pick list size of 30 products, than one in case of 2 products. It is because larger pick list have a greater probability of containing less popular SKUs, resulting in more travel to farther storage locations from home base point.

On the basis of Petersen, et. al. (2004), findings we assume the number of zones as three. It means that we will have three different storage classes. The zones are going to be dedicated for: fast moving products, medium moving products and slow moving products.

The only issue left to determine within this core is the percents of units realized by each of the zones, which is in literature called partition strategy. It will as a result determine the size of each zone. This is directly related to the demand skewness. In case of two zones the demand skewness might be for instance: 50-50, 30-70, 20-80. High demand skewness for instance 20-80 means that the top 20 % of top sale products account for 80 % of the total demand. Medium and low demand skewness means that 60 % and 40 % of the total demand for the 20 % of top sale products – Petersen II, C.G. (2000). For the three storage classes we might say about the 20-30-
50 partition, which means that top of 20 % of SKUs account for 80 % of the total demand, next 30 % of the top SKUs, accounts for 10 % of the total demand. The remaining 50 % of the products account for the 10 % of the total demand.

As we chose to use three storage classes, we have to analyze the data about the monthly sale of the products from Appendix 1. The comparison is done for each month separately, in order to establish average partition. From the analysis of the monthly data we might say that our partition is 10-15-75. It means that top of 10 % SKUs accounts for 80% of the total demand, next 15 % of the top of SKUs, accounts for 16 % of the demand. Remaining 75 % of the SKUs accounts for 4 % of the total demand. We might conclude that company represents high demand skewness – it means that couple of products are “doing” whole business, the rest of the products are only the background. The detailed distribution of product skewness might be found below in figure 5.1.

According to 10-15-75 there will be 140 fast moving products, 210 medium products and 1050 slow moving products. So we might see that the fast moving zone will be the smallest one, and the slow moving product zone will be the most area taking zone. Moreover, in the pick month the fast moving zone will have to deal with the products with average daily demands from 125 pieces up to almost 2500 pieces. The medium moving zone will have to support products with daily demands from average 20 pieces up to 125 pieces daily. Every daily demand below 4 pieces average daily will be supported by slow moving zone.

![Figure 5.1. Demand skewness.](image)
On the basis of demand skewness and the average order size from Section 2.1 we might assume that for the three zone system configuration there will be 4.8 SKU per average order to pick for fast moving zone, 0.98 SKU per average order to pick for middle zone and also 0.24 SKU per average order for the slow moving zone. In the system simulation to the mean value there will be added some deviation. In the result the SKU split might differ from order to order.

5.2 Station layout

The station layout section is going to be divided into three subsections, because of the number of storage classes. Each of the zones have different station layout and is going to be analyzed separately, due to differences in performance required. The main performance for the station is the picking time required to pick orders and the utilization of the picker. The station layout directly influences productivity of the picker. Hence productivity determines number of pickers required to serve particular amount of orders. In the reality this will be quite important performance since labor cost especially in developed countries is quite significant part in the costs structure.

In order to have basic data about the ergonomic work environment we will use the “Human dimensions atlas”, (2001). The atlas is used to determine the statistics considered with picking work – hands range dimension, one step range dimension. The atlas is also helpful to establish the working place dimensions – width of the paths. Each of the station layouts will be considered with the respect to the ergonomic rules.

5.2.1 Revision of the possible station arrangements

**Flow rack tunnel station**

It is station arrangement when there is a flow rack in front and behind the picker. The picker home base is situated in the middle of the tunnel. The advantage of this arrangement – comparing to reference arrangement, is that the picker does not have to walk so much. Instead there is a need to turn around in order to pick the item. Since the number of products being stored is the same for tunnel and referenced arrangement, the total horizontal travel distance is half in case of tunnel arrangement in comparison to the reference station arrangement. Another advantage of this type of station is so called “clear view”. It means that supervisor or group leader can stand in one point of line organized in the row and observed all the pickers simultaneously. Moreover the picking staff can help each other in case of lack of the line balance. The sample of the tunnel station arrangement is showed in figure 5.2.
The flow rack U shape station arrangement consists of one flow rack in the front of the picker and two on the sides. The main advantage of this arrangement is the amount of products available at the length of the arm, which is greater than in case of tunnel arrangement. In case of U shape arrangement, picker without moving from the home base of the station, neither turning around can easily reach 18 products, while in case of tunnel station or reference arrangement he/she has 6 products in hands area. Bigger amount of products in hands area give improvements in single picker productivity. The draw back of U shape arrangement is the lack of the clear view, which may cause lack of the temporary cooperation between pickers. Hence, the picker in order to pick the products from the back has to turn around by 180 degree, while in tunnel arrangement picker in order to pick from behind does not have to turn around by full 180 degree. The sample of the U shape station organization is depicted in figure 5.3.

This station configuration is showed in figure 4.2.(page 33) as the reference model for the slow moving products.
5.2.2 Fast moving products station layout

On the base of the split established in Section 5.1 the company has 140 fast moving products. The 140 top demand products are allocated to the fast moving segment. The average order size is on the base of assumption in Section 2.1 is described by poisson distribution with the mean value of 6 item. 80% of units is packed in the fast moving products zone. It is equal to 4,8 item per order. The average number of picks per referenced station within fast moving zone, since there are three stations in the raw will be described by the poisson distribution with mean value of 1,6.

If we take into consideration assumed in the reference model number of station in the raw and number of products stored in fast moving products zone, there should be 47 products per fast moving product station. The reference station layout is two floors, front flow racks arrangement. Within this arrangement picker has to pick the items only from the front of him/her. Detailed dimensions of the reference station are depicted in figure 5.4. The bin size should be equal to minimal dimension of maximal cartoon, in which the goods are packed in the factory. The bin high and width is assumed to be 40 cm, since the company profile which is interested in using system under consideration is cosmetic or pharmaceutical.

*Figure 5.4. The reference model station layout*
According to the *Human Dimensions Atlas, (2001)* the average women arms width is around 140 cm. In the reference model station layout, the picker has 6 bins in hands area – since there are two floors of bins. It means that the picker does not have to move from the home base location in order to make the pick activity. In one step area there is additionally 8 bins available. In figure 5.4., there are showed the particular areas and respectively the amount of bins. All the bins within the station are arranged into two floors. The picker always stays in front of the bin and the box.

In the tunnel arrangement there are 18 bins in two floors in front of the picker and 32 bins in 4 floors arrangement behind the picker. In order to pick from the behind the picker has to turn around. The advantage of above arrangement is the small area required for the station, mostly due to four floors back bins arrangement. The disadvantage is need from the picker to turn around in order to pick. This is not comfortable for the picker. Hence, two bins in exactly the same distance from the middle of the station, one on the front and one behind the picker, have different picking time due to necessity of turning around. Even more, picking the item from the front bin from the three steps area is less time taking than picking the item from the
back bin from two steps area. It states that turning around is time consuming activity. The tunnel station layout is showed in figure 5.5.

*Figure 5.6. U shape station layout.*

In the U shaped station layout the picker can pick the items without moving from the home base location from 18 bins, since front and one row of bins above conveyor is two floors bins and the raw just next to conveyor is 4 floors bins arrangement. The four floors are possible since we assumed the average high of the cartoon to be 40 cm. The main advantage is that the picker does not have to pick from the back, instead the picker picks from the side. This is more comfortable and less time consuming than picking from the back. The disadvantage of U shape station is lack of the clear view between stations. This might be seen as the advantage as well, since the lack of the direct contact with next picker allows avoiding the unnecessary contact between pickers.

**The model architecture**

The model architecture of one of the stations is depicted in figure 5.7. The missing two stations have exactly the same architecture and are connected just to the last assign module on figure 5.7. The detailed model documentation is shown in Appendix 5.
Figure 5.7. The 3 station in raw, U shape model architecture.

The description of the modules from figure 5.7 is as follows:

1. Create module
   The module is responsible for entity creation. The entity in out model states for the box with the order.

2. Assign module
   The module is responsible for assigning to a variable, which describes the number of queuing orders, orders which are going into the station. The module every time when the entity is passing the through adds one to the present value of variable.

3. Decide module
   The module is responsible for control of picker number one busy/idle state. We have couple of processes that picker has to perform sequentially and cannot begin packing next order without ending previous one. The module checks the value of defined variable. The variable might have value 0 or 1. The 0 value means that there is no order in the queue and picker is idle; 1 means that there is a queue and picker is busy.
4 Hold module

The aim of the module is to store the order until the signal will appear. The implementation of this module is necessary since we have couple of processes that picker has to perform sequentially. The hold module stores the order until the signal value from the module signal (16) will appear.

5 Assign module

The module is responsible for assigning to the entity – order couple of variables. The module assigns variable Licznik1 value of 0. The variable will be further used to count the number of products picked within the station. The module assigns to the entity variable Repeat1 value anint(poiss(1,6)). Within this variable we determine the number of items being picked from the station number one. The module assigns to the entity variable 2 minus one value. The variable is responsible for subtracting served order from the queue. Finally the modules assigns to the entity variable 1, with the value 1. This provides information that the picker 1 becomes busy.

6 Process module

Process simulates activity of picking reading the picking list from the box. The process is assumed to have triangular distribution. The most likely value of the process is 1 second on the base of Hinojosa, Arturo (2003)

7 Decide module

The module is responsible for assigning the order to the particular group of the bins. The bins were divided on the base of areas of availability. Percentage value of each of the branches of the decide module are based on percentage participation of this
group into the whole volume of the station. The list of the products that should be stored within first station we have created by taking every third product sale from the fast moving zone – detailed data in Appendix 1. Then we divided them into the groups on the base of station layout 5.6, being aware the availability areas. Afterward we have taken the percentage values. The vertical bins location we have assumed to be negligible. The module simulates as well the orders that do not have anything to be picked within the station. In case of such orders the picker just picks the picking list.

8 Process module

The module is responsible for simulating the picking activity within hands area. The picker does not have to move from the home base point. We assume value of 0.2 seconds.

9 Process module

The module is responsible for simulating the picking activity within one step area. The picker has to move one step from the home base point in order to make a pick. The mean value was taken on the base of Hinojosa, Arturo (2003).
10 Process module

The module is responsible for simulating the picking activity within two step area. The picker has to move two steps from the home base point in order to make a pick. The mean value was taken on the base of Hinojosa, Arturo (2003).

11 Process module

The module is responsible for simulating the turning around activity. The picker has to turn around in order to make a pick from the one and two steps area. We have assumed the mean value of 2 seconds.

12 Process module

The module is responsible for simulating the dropping items into the box and releasing the box activity. The picker has to drop the items into the order and release the box. We have assumed the mean value of 1,4 seconds.
13 Record module
The module is responsible for recording entity statistics.

14 Assign module
The module is responsible for assigning to the entity variable Licznik 1 new value of Licznik 1+1. The variable is necessary in order to control the number of items that have been already picked from the station. Every time when the entity passes the module it means that another item had been picked.

15 Decide module
The module is responsible for checking if the order has packed within the station desired number of items. If desired number of items (Repeat 1 variable) has been packed, the order is released to next station. If desired number of items has not been packed yet the order is released once again to picking processes.

16 Signal module
The module is responsible for sending the signal to hold module, which will release next order to be processed.

17 Assign module
The module is responsible for assigning to the variable 1 new value 0. The variable 1 is used by decide module described in point 3 in order to check the state of the picker. The module assigns as well variable 4, which is used by next station in the way described in point 2.

The other shapes of stations were created by changing the percentage values of decide module from point 7. In case of reference station layout there was a need to add additional three processes, which simulate bins at three, four and five steps area.
The second station was created by copying station layout from figure 5.7, and adding it at the end of first station. The detailed model documentation might be found in Appendix 5.

**The simulation results**

For each of the station layout, simulation was run for the time equal to the two shifts work – 15 EWH. In order to avoid the influence of the statistical disturbances each simulation was run 10 times, which is equal to ten working days. This provides the simulation credibility. The warm up period was not implemented since in reality when the work starts it should start within the empty system. Each picker has the same performance, including the failure rate. We simulate the system within the option that in average 4% of boxes does not include any item to pick from the particular station. According to the reference model there are three stations in the raw. The simulation was run for the three stations in the raw arrangement. The results of the experiment are presented in Table 2.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The reference model</th>
<th>The tunnel</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>9.6</td>
<td>9.6</td>
<td>9.6</td>
</tr>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>5,519</td>
<td>5,624</td>
<td>5,593</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>27725</td>
<td>28253</td>
<td>28097</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>506.21</td>
<td>82.58</td>
<td>60.00</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>8.67</td>
<td>4.80</td>
<td>4.36</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>1490.86</td>
<td>397.68</td>
<td>275.99</td>
</tr>
<tr>
<td>Average picking time</td>
<td>27.87</td>
<td>23.01</td>
<td>20.80</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>2.68</td>
<td>2.91</td>
<td>3.05</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>101.19</td>
<td>74.53</td>
<td>72.81</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>478.33</td>
<td>59.57</td>
<td>39.20</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>1454.39</td>
<td>373.63</td>
<td>260.81</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>95.80%</td>
<td>80.12%</td>
<td>71.73%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>94.76%</td>
<td>79.44%</td>
<td>71.12%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>95.11%</td>
<td>80.19%</td>
<td>72.72%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>202.00</td>
<td>30.36</td>
<td>20.32</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>874.59</td>
<td>268.08</td>
<td>224.37</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>138.42</td>
<td>23.08</td>
<td>17.22</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>821.52</td>
<td>148.46</td>
<td>147.77</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>158.95</td>
<td>22.87</td>
<td>18.13</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>945.27</td>
<td>214.59</td>
<td>198.99</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>20.34</td>
<td>2.57</td>
<td>1.53</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>97.00</td>
<td>34.00</td>
<td>26.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>13.58</td>
<td>1.83</td>
<td>1.21</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>86.00</td>
<td>19.00</td>
<td>18.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>15.55</td>
<td>1.81</td>
<td>1.31</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>99.00</td>
<td>26.00</td>
<td>22.00</td>
</tr>
</tbody>
</table>

Table 2. The simulation results for the fast moving products.

From table 2 we can see that the highest number of orders is served when the station is in tunnel arrangement. Tunnel arrangement serves 0.5% more orders than the U shape station, and 1.5% more than referenced arrangement. When considering average times, the best from considered solutions is U shape station. U shape arrangement offers 25% reduction of average time comparing to tunnel station, and 88% comparing to reference arrangement. Most of the saving is done by reduction of waiting times. The picking time in case of U shape arrangement is reduced by 26% comparing to referenced solution. In case of tunnel arrangement, the picking times are reduced by 16% comparing to referenced solution. Moreover, utilization of single picker drops from 95% in referenced solution to 71% in case of U shape station. The
waiting times are the result of the number of orders in the queues, which is changing in the same way as the waiting times.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time. The results are depicted in Table 3.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>The reference model</th>
<th>The tunnel</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>9.6</td>
<td>8.1</td>
<td>7.2</td>
</tr>
</tbody>
</table>

**General**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>The reference model</th>
<th>The tunnel</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orders out</td>
<td>5 519</td>
<td>6 642</td>
<td>7 421</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>27725</td>
<td>33 353</td>
<td>37 567</td>
</tr>
</tbody>
</table>

**Times (seconds)**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>The reference model</th>
<th>The tunnel</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average total time (left scale)</td>
<td>506.21</td>
<td>282.46</td>
<td>374.65</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>8.67</td>
<td>7.84</td>
<td>8.16</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>1490.86</td>
<td>911.15</td>
<td>1000.67</td>
</tr>
<tr>
<td>Average picking time</td>
<td>27.87</td>
<td>23.06</td>
<td>20.80</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>2.68</td>
<td>2.19</td>
<td>2.41</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>101.19</td>
<td>83.02</td>
<td>71.10</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>478.33</td>
<td>259.41</td>
<td>353.85</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>1454.39</td>
<td>894.27</td>
<td>986.37</td>
</tr>
</tbody>
</table>

**Resources**

| Picker 1 utilization                          | 95.80%              | 94.82%     | 95.37%      |
| Picker 2 utilization                          | 94.76%              | 94.55%     | 94.64%      |
| Picker 3 utilization                          | 95.11%              | 94.63%     | 96.44%      |

**Queues times (seconds)**

| Picker 1 average queuing time                 | 202.00              | 114.69     | 125.99      |
| Picker 1 minimum queuing time                 | 0.00                | 0.00       | 0.00        |
| Picker 1 maximum queuing time                 | 874.59              | 622.74     | 770.18      |
| Picker 2 average queuing time                 | 138.42              | 86.22      | 86.10       |
| Picker 2 minimum queuing time                 | 0.00                | 0.00       | 0.00        |
| Picker 2 maximum queuing time                 | 821.52              | 554.60     | 396.89      |
| Picker 3 average queuing time                 | 159.95              | 72.97      | 157.02      |
| Picker 3 minimum queuing time                 | 0.00                | 0.00       | 0.00        |
| Picker 3 maximum queuing time                 | 945.27              | 644.81     | 732.73      |

**Number in queues**

| Picker 1 average queuing number                | 20.34               | 13.62      | 16.81       |
| Picker 1 minimum queuing number                | 0.00                | 0.00       | 0.00        |
| Picker 1 maximum queuing number                | 97.00               | 78.00      | 104.00      |
| Picker 2 average queuing number                | 13.58               | 10.08      | 11.20       |
| Picker 2 minimum queuing number                | 0.00                | 0.00       | 0.00        |
| Picker 2 maximum queuing number                | 86.00               | 78.00      | 59.00       |
| Picker 3 average queuing number                | 15.55               | 8.40       | 20.94       |
| Picker 3 minimum queuing number                | 0.00                | 0.00       | 0.00        |
| Picker 3 maximum queuing number                | 99.00               | 82.00      | 97.00       |

Table 3. The simulation results for the fast moving products.

The results depict without any doubt that the U shape layout is the most appropriate from considered for the fast moving products. With the U shape station the total time that each order spends in the system is 10 % higher than in case of tunnel station arrangement, but it is still 25 % lower than in case of the reference model. Total
number of orders out is higher than in case of tunnel layout for around 10%. The units served are also greater for the U shape station. This will mean ten percents rise in productivity in case of U shape arrangement. Hence, looking at the utilization of the pickers, we see that still there is field for the improvements in case of tunnel station. The U shape layout has similar utilization rates as the reference model, but still there is a place for improvements. The productivity is the lowest for the reference model. It is explained by the long walking routes within station. This makes picking the item time taking activity. Better performance of the U shape model than the tunnel model might be explained as the result of lower variations in number of queuing orders. Due to limitations of Arena Academic Version, the system is constrained by the quantity of the orders, which determines maximum length of the queues. The picking system was limited by the 150 entities – the number of the orders in the system can not exceed the number 150. It is reasonable since the space on the conveyor is not infinite, and building huge buffers is not the solution. For the order picking systems the important factor is the time spends in the system – since most of the companies guarantee the shipment within short period of time. In case of referenced model the number of working process is the highest – the highest average number of items in the queues. The tunnel shape station has the lowest number in the queue. The queue length is the highest in case of the picker number one and the lowest for the third picker. This situation suggests that in practice when assigning the pickers to the stations we should assign the fastest picker to the first station. It will improve performance of all stations put in line. In case of the U shape station layout, the lowest number of orders waiting in the line is for the second picker. The lowest average number of items in the queue might be explained as the statistical distribution disturbances. There is higher variety in between arrival times, which in the result affect the number of items in the queue.

The best solution from considered for the picking activity in the fast moving products zone is the U shape station layout. According to Appendix 1, the fast moving zone needs to be able to pack 56000 units per day (1113000 per month). Within U shape arrangement it is possible with two lines. Two lines are also required within tunnel arrangement, but in that case the buffer is lower. The reference solution requires three lines in order to serve monthly demand, which means hiring nine persons instead six in case of rest of the solutions.
5.2.3 Number of fast moving products U shape stations.

We will widen the analysis on the number of stations in the raw for the best solution from Subsection 5.2.4. The scenarios examined will include 1, 2, 3 or 4 stations put in the raw. We do the analysis of the best from considered number of stations in the line for the station layout chosen in Subsection 5.2.2. We maximize the benefits by adjusting the number of stations in the raw. The simulations conditions are exactly the same as for the simulation from Subsection 5.2.4. — simulation was run for the time equivalent to the 2 shifts work for ten days. Afterwards the average performances were count. Each of the stations is served by one picker.

The model architecture

The model is similar to the model architecture from Subsection 5.2.2. The only difference is in percentages values in decide module from point 7. The percentage values differ because the fast moving products zone sale values are divided now between 4 stations and not as in the previous section between 3 stations.

The simulation results

The simulation showed that the greater number of stations in the raw the better critical system performance. The results are given in Table 4.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>32.3</td>
</tr>
<tr>
<td></td>
<td>32.3</td>
</tr>
<tr>
<td></td>
<td>32.3</td>
</tr>
<tr>
<td></td>
<td>32.3</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>1598</td>
</tr>
<tr>
<td></td>
<td>1689</td>
</tr>
<tr>
<td></td>
<td>1637</td>
</tr>
<tr>
<td></td>
<td>1663</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>6753</td>
</tr>
<tr>
<td></td>
<td>7138</td>
</tr>
<tr>
<td></td>
<td>6920</td>
</tr>
<tr>
<td></td>
<td>7027</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>1437.18</td>
</tr>
<tr>
<td></td>
<td>33.78</td>
</tr>
<tr>
<td></td>
<td>27.80</td>
</tr>
<tr>
<td></td>
<td>23.61</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td>4.24</td>
</tr>
<tr>
<td></td>
<td>5.09</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>3977.21</td>
</tr>
<tr>
<td></td>
<td>176.24</td>
</tr>
<tr>
<td></td>
<td>148.44</td>
</tr>
<tr>
<td></td>
<td>176.46</td>
</tr>
<tr>
<td>Average picking time</td>
<td>33.37</td>
</tr>
<tr>
<td></td>
<td>23.29</td>
</tr>
<tr>
<td></td>
<td>23.02</td>
</tr>
<tr>
<td></td>
<td>20.29</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>1.39</td>
</tr>
<tr>
<td></td>
<td>2.88</td>
</tr>
<tr>
<td></td>
<td>5.09</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>167.21</td>
</tr>
<tr>
<td></td>
<td>83.48</td>
</tr>
<tr>
<td></td>
<td>86.24</td>
</tr>
<tr>
<td></td>
<td>68.42</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>1403.81</td>
</tr>
<tr>
<td></td>
<td>10.49</td>
</tr>
<tr>
<td></td>
<td>4.78</td>
</tr>
<tr>
<td></td>
<td>3.32</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>3942.32</td>
</tr>
<tr>
<td></td>
<td>152.96</td>
</tr>
<tr>
<td></td>
<td>121.37</td>
</tr>
<tr>
<td></td>
<td>147.56</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>98.74%</td>
</tr>
<tr>
<td></td>
<td>36.23%</td>
</tr>
<tr>
<td></td>
<td>23.14%</td>
</tr>
<tr>
<td></td>
<td>15.45%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>36.66%</td>
</tr>
<tr>
<td></td>
<td>23.27%</td>
</tr>
<tr>
<td></td>
<td>15.50%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>23.41%</td>
</tr>
<tr>
<td></td>
<td>15.74%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>15.77%</td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>1411.19</td>
</tr>
<tr>
<td></td>
<td>15.20</td>
</tr>
<tr>
<td></td>
<td>7.57</td>
</tr>
<tr>
<td></td>
<td>4.76</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>3942.32</td>
</tr>
<tr>
<td></td>
<td>102.97</td>
</tr>
<tr>
<td></td>
<td>61.46</td>
</tr>
<tr>
<td></td>
<td>78.88</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>13.84</td>
</tr>
<tr>
<td></td>
<td>7.16</td>
</tr>
<tr>
<td></td>
<td>4.83</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>128.05</td>
</tr>
<tr>
<td></td>
<td>100.11</td>
</tr>
<tr>
<td></td>
<td>82.75</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>6.68</td>
</tr>
<tr>
<td></td>
<td>4.95</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>44.29</td>
</tr>
<tr>
<td></td>
<td>80.44</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5.26</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>128.07</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>43.23</td>
</tr>
<tr>
<td></td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>120.00</td>
</tr>
<tr>
<td></td>
<td>8.00</td>
</tr>
<tr>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>7.00</td>
</tr>
</tbody>
</table>

Table 4: The simulation result for the fast moving products for the U shape station for various number of stations in the raw.

From table 4 we can see that similar number of orders is served by each of the solutions except one station in raw arrangement. One station arrangement serves around 3% less orders those other solutions. The differences are appearing in average
total time, which is the smallest for 4 stations in raw arrangement. The reduction of total time is achieved due to dropping picking and waiting times. Moreover, single picker utilization is the smallest in case of 4 station arrangement, what suggests that resources are not properly used.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time.

The critical system performances are depicted in figure 5.8. The average total time is the time that average order spends in the fast moving products zone. Basically it says that in average order spends 1400 seconds in the single station system. This value is lower in case of 4 stations arrangement. The average order spends almost 10% of this value in the four stations system. Most of these times are consumed by queuing in lines to the picker. Especially in case of one station arrangement the box spends in the queue almost 97% of the total time in the system. Relatively this value drops to the 89% in case of 4 station arrangement.

![Figure 5.8. The performance summary.](image)

The aim of our analysis is to choose the solution which minimizes the number of staff needed and length of the conveyor. The average number in queue plus standard deviation in connection with the dimensions of the order box should give us the knowledge about the length of the conveyor between the stations. The queue lengths are depicted in figure 5.9. Setting up the length of the conveyor should be a trade off between optimal usage of resources and line balance issues.
Figure 5.9. The average number of queuing orders.

Hence, there is a need to efficiently use the resources – pickers, by minimizing the idle times. In the system it means maintaining continuous supply of the orders to the picker. This performance is highlighted in figure 5.10., in which we summarize the average picker utilization. The figure shows that lower number of stations in the raw means higher single picker utilization. The exception is the layout with the two stations in the raw. Within this arrangement both pickers have similar utilization to average picker in three station arrangement. In two station arrangement there is also very small difference in both pickers utilization – each of them is utilized in around 95.1%.

There is a possibility to extend the single picker utilization, by rising up the number of orders in the line system. Within the simulation the total number of the orders was limited by 150 boxes – Arena limitations. It is reasonably in reality due to limited capacity of buffers. The rising up the number of boxes in the system lead to the higher numbers in the queue – average and maximal. Higher numbers in the queue will lead to greater times spend in the queues and as a consequence higher total times. Expanding the number of boxes in the system will raise number of working in process.
Figure 5.10. The average picker utilization

The utilization of the single picker is influenced by the line balance and statistical disturbances issues. The line balance considers the workload of the single picker. It is very hard to provide well functioning arrangement of stations in line. Even if we provide extremely equal workload for each of the stations, the single picker performance make very hard to reach situation of perfect balance between single picker performance and station workload. The perfect balance situation is described as a situation under each to each picker there are the same picking times and number of orders in the queue. In reality it is extremely difficult to arrange the match pickers to stations in order to provide well balanced line. For the simulation needs the methodology of assigning product to the bins in case of three station arrangement was as follows: the highest sale item was assigned to the closest to the picker bin on station one; the second largest sale product was assigned to the same bin on station two; the third product to the closest to the picker bin on station three; the fourth sale product was assigned to the second closest bin on the station one; and so on. The procedure was run until all the products were assigned. Within this procedure the first picker is the one that has the most pick to perform, due to assigning always maximal sale number product. When we take a look on all the pickers working on the first stations, the workload that they have to serve is the highest one. This would intend once again that the picker one should be the fastest one. In is possible to implement different logic for assigning the products to the stations. We can for instance assign first top sale product to the first station, then second top sale product to the second station, third top sale to third station, the fourth one to the fourth station; and then the fifth top sale product to the fourth station, sixth one to the third station, seventh one to the second station, and the eighth top sale product to the first station, and so on. Such arrangement will provide more equal workload especially between stations one and four.
On the other side the picker number one makes core value of the picks in the closest area to the home base – when we compare the percentages of picks performed from particular bins area. Comparing to the picker number four, she / he has better conditions for the picking. The workload of each station is showed in the figure 5.11. The higher number of stations the more visible differences between the workload and as the result troubles with line balance. We develop further this issue in the Section 5.3 – storage policies, in order to gather best from considered pickers’ performance. The differences in percents workload to the area of picking within the station are showed in figure 5.12. First picker in all cases has more percents of picks in the hands area, due to locating within this area always the top products. The first picker always receives the highest sale value products, and in the result the workload of first picker is the highest.

Hence the factor which influences the single picker utilization is the statistical disturbances. On each station, since we assumed in line arrangement, times between arrivals are described by different statistical function. The order arrives to the first station with the exponential distribution with the mean value, and then it is passed to the second picker. Order arrives to the second picker with the distribution which is the result of the exponential distribution of between arrival times and the triangular picking times of the first picker. When the same order arrives to the picker number three it is disrupt by the exponential between arrival times and the triangular picking times of the first and second picker. In the result the fourth picker receives the orders with very high comparing to the first picker amplitude arrivals times. Higher amplitude means bigger chance for not continuous supply of the orders to the picker – the idle times in the process. In such a case the picker utilization will drop. The last picker is always the one which is influenced the most within this issue that is why its utilization might be lower than the average.

Figure 5.11. Comparison of station workload
From the figure 5.11 we see that in case of four station arrangement the highest value of workload is served by first station. This would make the first picker busier than the other pickers. The capacity of the single picker is also influenced by the within station work split depicted in figure 5.12. The first picker comparing to the other pickers has more pick to perform within the best picking area – hands area. This should slightly compensate the higher units load. Still it is reasonably that the first picker is the fastest one.

The simulation was run for ten working days. The daily hour limits was equal to the two shifts work scheduled – 15 working hours. Each picker was simulated with considering separately the failure rates, which in reality correspond for instance to time when picker talks with other picker. The summary of the results are depicted in table 5.
### Statistic

<table>
<thead>
<tr>
<th>Stations in the raw</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>1</td>
</tr>
<tr>
<td>32,3</td>
<td>12,2</td>
</tr>
</tbody>
</table>

#### General

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orders out</td>
<td>1,598</td>
<td>4,392</td>
<td>7,421</td>
<td>9,787</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>6,753</td>
<td>20,090</td>
<td>37,567</td>
<td>55,372</td>
</tr>
</tbody>
</table>

#### Times (seconds)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average total time (left scale)</td>
<td>1,437.18</td>
<td>358.48</td>
<td>374.65</td>
<td>185.91</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>0.96</td>
<td>3.18</td>
<td>8.16</td>
<td>7.56</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>3,977.21</td>
<td>1,231.24</td>
<td>1,000.67</td>
<td>642.39</td>
</tr>
<tr>
<td>Average picking time</td>
<td>33.37</td>
<td>23.39</td>
<td>20.80</td>
<td>20.31</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>0.52</td>
<td>1.37</td>
<td>2.41</td>
<td>4.25</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>167.21</td>
<td>89.87</td>
<td>71.10</td>
<td>81.40</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>1,403.81</td>
<td>335.09</td>
<td>353.85</td>
<td>165.60</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>3,942.32</td>
<td>1,215.44</td>
<td>986.37</td>
<td>625.16</td>
</tr>
</tbody>
</table>

#### Resources

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 utilization</td>
<td>98.74%</td>
<td>95.35%</td>
<td>95.37%</td>
<td>91.02%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>-</td>
<td>95.31%</td>
<td>94.64%</td>
<td>91.73%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>-</td>
<td>-</td>
<td>96.44%</td>
<td>93.10%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>92.64%</td>
</tr>
</tbody>
</table>

#### Queues times (seconds)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 average queuing time</td>
<td>1,411.19</td>
<td>183.79</td>
<td>125.99</td>
<td>46.28</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.47</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>3,942.32</td>
<td>38.36</td>
<td>770.18</td>
<td>433.73</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>-</td>
<td>167.10</td>
<td>86.10</td>
<td>38.40</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>-</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>-</td>
<td>69.64</td>
<td>396.89</td>
<td>254.94</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>-</td>
<td>-</td>
<td>157.02</td>
<td>49.75</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>-</td>
<td>-</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>-</td>
<td>-</td>
<td>732.73</td>
<td>314.44</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>46.59</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>346.34</td>
</tr>
</tbody>
</table>

#### Number in queues

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 average queuing number</td>
<td>43.23</td>
<td>14.51</td>
<td>16.81</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>120.00</td>
<td>73.00</td>
<td>104.00</td>
<td>88.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>-</td>
<td>12.95</td>
<td>11.20</td>
<td>6.30</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>-</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>-</td>
<td>67.00</td>
<td>59.00</td>
<td>48.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>-</td>
<td>-</td>
<td>20.94</td>
<td>8.31</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>-</td>
<td>-</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>-</td>
<td>-</td>
<td>97.00</td>
<td>61.00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>66.00</td>
</tr>
</tbody>
</table>

**Table 5.** The simulation result for the fast moving products for the U shape station for various number of stations in the raw.

We can see that the U shaped station with four station arrangement will perform the highest number of orders and units out. Also single number in the queue is smaller than in other cases, which is explained as the result of reduced travel distances within
the station. Due to the smaller size of the station we achieve reduction in time needed for travel within the station. Hence the utilization might be improved through increasing the number of orders in the system – currently 150, and by changing the line balance to the more equal one. The higher number of items in the system will cause for us higher average and maximal number of orders in the queue. In the result there will be more space for the buffer queues required – more space on the conveyor.

The fast moving products zone is required to serve 80% of units - according to the units per zone split. On the base of data in Appendix 1 it is around 56000 units per day. The average number of units and orders served is varying depends on the number of stations in the line. The data are depicted in table 5. The fast moving products zone need to pack 56000 units per day. It is possible to pack this quantity of units with nine lines within one station in the raw, which means hiring nine picking persons. The same amount of units might be pack with three lines of two stations in the raw. This solution requires six pickers. In order to serve the demand within three stations in the raw, two lines are required – six pickers. Respectively, the same situation requires almost only one line of the four stations in the raw arrangement. Since our goal is to minimize number of staff hired, the best from considered solutions for the fast moving products zone is U shape four stations in raw arrangement.

5.2.4 Medium moving products station layout

On the base of the split in Section 5.1 the company has 210 medium sale products. The 210 middle demand products are allocated to the medium moving segment. The average order size is on the base of assumption in Section 2.1 is described by poisson distribution with the mean value of 6 item. On the base of units zone split, 16% of this value is done by the medium moving products zone. It is equal to 0.92 item per order. The average number of items per referenced station within medium moving zone will be described by the poisson distribution with mean value of 0.306, since we have 3 stations in raw. The medium moving zone has to pack 16% of total units, which are 11200 units per day.

Since we did not developed medium moving products zone in reference model, we examined the medium moving products zone the same way as we examined fast moving products zone. Reference fast moving station arrangement consists of three stations in the line. Each station of medium moving products zone contains bins for 70 products. Items are located in two floor arrangement of bins, on the flow racks. The argumentation for using the flow racks is that within the medium moving products it is reasonable from the replenishment point of view to store couple of cartons simultaneously. This is not possible in case of static shelves, in which single bin has the capacity of one carton. The bins dimensions are exactly the same as in case of fast moving zone – the width of 40 cm. The picker has all the bins in front of him/her. There is no station bypass, nor zone one.
The stations layouts are analogous to those for the fast moving products zone. In case of reference model, in order to have 70 bins we have to add 10 bins on each side of the station. The bins are arranged in two floors. The bins are added as the continuation for the bins arrangement from the figure 5.4 (page 40).

In case of tunnel arrangement, in order to hold all the products within the station there is a need to add on each side four bins in the front and eight bins at the back. The bins in the front are arranged in two floors, and those on the back into four floors. The bins are added as the continuation for the bins arrangement from the figure 5.5 (page 41).

Since 3 stations in line arrangement in fast moving products zone has place to store 47 products per station and under the same arrangement in medium moving products zone we store 70 products per station, for the U shape station arrangement there is a need to add to the sides of the station twelve bins on each side. The bins are added to the end of the station, which is in the opposite point to the picker home base location. The bins are added in the four floors arrangement. The bins are added at the bottom of the station on the figure 5.6 (page 42).

**The model architecture**

The model is similar to the model architecture from Subsection 5.2.2. There is only one difference. It is in percentages values in decide module from point 7, which is result of different sale split. The rest of the model architecture stays unchanged to the model from Subsection 5.2.2.

**The simulation results**

The simulation conditions are identical as the fast moving products zone simulation. For each of the station layout simulation were run for the time equal to the two shifts work – 15 effective working hours. In order to avoid the influence of the statistical disturbances each simulation were run 10 times, which is equal to ten working days. This makes the simulation credibility. The warm up period was not implemented since in reality when the work starts it should start within the empty system. Each picker has the same performance, including the failure rate. Since there is no bypass, we simulate the system within an option that average 4 % of boxes does not include any item to pick from the particular station. According to the reference model there are three stations in the raw. The results of the experiment are presented in Table 6.
# Table 6. The simulation results for the medium moving products.

From table 6 we can see that each of considered solutions serves similar number of orders. When considering average times, the best from considered solutions is tunnel arrangement. Tunnel arrangement offers 10 % reduction in average total time comparing to the U shape station and 76 % reduction comparing to reference arrangement. The waiting times are reduced in proportion to average total time. Moreover, utilization of single picker drops from 83% in referenced solution to 63% in case of tunnel and U shape station. The waiting times are the result of the number of orders in the queues, which is changing in the same way as the waiting times.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time. The results are depicted in Table 7.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>The reference model</th>
<th>The tunnel</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>3.8</td>
<td>3.8</td>
<td>3.8</td>
</tr>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>14 181</td>
<td>14 199</td>
<td>14 190</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>13 029</td>
<td>13 046</td>
<td>13 037</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>106.27</td>
<td>24.72</td>
<td>27.74</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>2.17</td>
<td>1.86</td>
<td>1.93</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>390.02</td>
<td>192.60</td>
<td>168.29</td>
</tr>
<tr>
<td>Average picking time</td>
<td>9.52</td>
<td>7.21</td>
<td>7.27</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>1.73</td>
<td>1.75</td>
<td>1.70</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>89.40</td>
<td>51.73</td>
<td>52.25</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>96.75</td>
<td>17.51</td>
<td>20.47</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>386.05</td>
<td>189.17</td>
<td>164.93</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>83.97%</td>
<td>63.21%</td>
<td>63.75%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>83.34%</td>
<td>63.05%</td>
<td>63.33%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>82.92%</td>
<td>63.43%</td>
<td>63.90%</td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>34.31</td>
<td>8.50</td>
<td>9.72</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>316.33</td>
<td>135.93</td>
<td>108.40</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>37.68</td>
<td>8.94</td>
<td>10.16</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>297.53</td>
<td>104.07</td>
<td>148.73</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>38.86</td>
<td>9.09</td>
<td>10.51</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>265.54</td>
<td>125.20</td>
<td>140.98</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>7.65</td>
<td>1.43</td>
<td>1.65</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>77.00</td>
<td>40.00</td>
<td>35.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>8.73</td>
<td>1.55</td>
<td>1.80</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>90.00</td>
<td>33.00</td>
<td>49.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>9.06</td>
<td>1.61</td>
<td>1.92</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>72.00</td>
<td>37.00</td>
<td>36.00</td>
</tr>
</tbody>
</table>
Table 7. The simulation results for the medium moving products.

The simulation results show that the U shape and tunnel station are much better than the reference arrangement. The U shape arrangement is little bit better than the tunnel station, if we take a look at the total number of orders out. The U shape station serves within the same time 3% more orders than the tunnel station. The U shape station is the best if it goes about the main performance of the system – number of orders out, but if we take a look at other performances the leading solution becomes the tunnel arrangement. The main drawback of the U shape station for medium moving product is queue size. The numbers of items in the line and queuing times are the highest from all solutions. In the result average order spends the longest time in the queue. This extends the number of working processes, and forces us to build up buffers before each station. For U shape station buffers will be the biggest one, and in average should...
have a space to accumulate 14 queuing boxes. Longer line length has also advantage – it allows for better resource utilization. Each picker in case of U shape arrangement has much higher utilization than for the other solutions. The U shape arrangement, due to bigger buffers, provides more continues string of the work to be done. The picker has less idle time. Even in case of U shape station, still there is place to improve single picker utilization. The one of the possible solutions will be some bypass which will eliminate boxes with no pick from particular station. This solution should also extend the productivity of whole system. Hence for our simulation there is constraining that the maximum number of orders can not extend 150 – version of Arena. If we take a look on time between arrivals values, there difference in only 0.1 second. In the tunnel station, 2.7 seconds is the minimal time between arrivals. Due to limitations of academic version of Arena, if we implement lower time between arrivals we extend 150 orders in the system during simulation. The same arrival time for the U shape station does not cause any problems. It suggests that the tunnel station is more sensitive for the time between arrivals values, and it easier get stack. It stays in contradiction with lower average number of items in queue, which is lower for the tunnel arrangement. It shows that the tunnel station is more sensitive for the changes in times between arrivals.

From the results summary in table 7 we may say that the best solution from considered is U shape station – as long as the space is not constraining us. If the space on the conveyor becomes crucial factor we should choose the tunnel arrangement. We assumed that the space is not constraining us, so the best solution from considered for the medium moving products segment will be the U shape station.

5.2.5 Number of medium moving products stations.

We will widen the analysis on the number of station in the raw for the best solution from Subsection 5.2.4. The scenarios examined will include 1, 2, 3 or 4 stations put in the raw. The simulations conditions are exactly the same as for the simulation from Subsection 5.2.4. – the simulation was run for the time equivalent to the 2 shifts work for ten days. Afterwards the average performances were count.

The model architecture

The model is similar to the model architecture from Subsection 5.2.3. There is only one difference. It is in percentages values in decide module from point 7, which is result of different sale split. The rest of the model architecture stays unchanged to the model from Subsection 5.2.3.

The simulation results

Once again the simulation showed that the greater number of stations in the raw the better critical system performance. We simulate each of the models with the same between arrival times. The results are depicted in Table 8.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td></td>
</tr>
<tr>
<td>11,8</td>
<td>11,8</td>
</tr>
<tr>
<td>11,8</td>
<td>11,8</td>
</tr>
<tr>
<td>11,8</td>
<td>11,8</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td></td>
</tr>
<tr>
<td>4512</td>
<td>4551</td>
</tr>
<tr>
<td>4548</td>
<td>4582</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td></td>
</tr>
<tr>
<td>4151</td>
<td>4187</td>
</tr>
<tr>
<td>4184</td>
<td>4216</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td></td>
</tr>
<tr>
<td>284,79</td>
<td>13,93</td>
</tr>
<tr>
<td>10,47</td>
<td>9,99</td>
</tr>
<tr>
<td>Minimum total time</td>
<td></td>
</tr>
<tr>
<td>0,53</td>
<td>1,09</td>
</tr>
<tr>
<td>1,74</td>
<td>2,58</td>
</tr>
<tr>
<td>Maximum total time</td>
<td></td>
</tr>
<tr>
<td>1203,25</td>
<td>152,29</td>
</tr>
<tr>
<td>121,66</td>
<td>155,17</td>
</tr>
<tr>
<td>Average picking time</td>
<td></td>
</tr>
<tr>
<td>11,32</td>
<td>7,82</td>
</tr>
<tr>
<td>7,28</td>
<td>7,46</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td></td>
</tr>
<tr>
<td>0,51</td>
<td>1,09</td>
</tr>
<tr>
<td>1,74</td>
<td>2,52</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td></td>
</tr>
<tr>
<td>101,92</td>
<td>63,92</td>
</tr>
<tr>
<td>48,49</td>
<td>42,43</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td></td>
</tr>
<tr>
<td>273,47</td>
<td>6,11</td>
</tr>
<tr>
<td>3,20</td>
<td>2,53</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td></td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td></td>
</tr>
<tr>
<td>1200,56</td>
<td>141,55</td>
</tr>
<tr>
<td>103,58</td>
<td>146,03</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td></td>
</tr>
<tr>
<td>94,64%</td>
<td>32,91%</td>
</tr>
<tr>
<td>20,53%</td>
<td>15,87%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>32,97%</td>
</tr>
<tr>
<td>20,40%</td>
<td>15,86%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>20,36%</td>
<td>15,83%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>15,75%</td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td></td>
</tr>
<tr>
<td>285,45</td>
<td>8,38</td>
</tr>
<tr>
<td>4,47</td>
<td>3,38</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>1200,56</td>
<td>73,52</td>
</tr>
<tr>
<td>98,14</td>
<td>144,18</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>8,72</td>
</tr>
<tr>
<td>4,61</td>
<td>3,18</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0,00</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>132,65</td>
</tr>
<tr>
<td>93,17</td>
<td>102,71</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4,68</td>
<td>3,51</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>88,05</td>
<td>129,45</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>3,69</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>131,29</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td></td>
</tr>
<tr>
<td>23,01</td>
<td>0,24</td>
</tr>
<tr>
<td>0,08</td>
<td>0,05</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>103,00</td>
<td>10,00</td>
</tr>
<tr>
<td>11,00</td>
<td>15,00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0,27</td>
</tr>
<tr>
<td>0,08</td>
<td>0,04</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0,00</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>14,00</td>
</tr>
<tr>
<td>11,00</td>
<td>14,00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0,09</td>
<td>0,05</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10,00</td>
<td>12,00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>0,05</td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>17,00</td>
</tr>
</tbody>
</table>

Table 8. The simulation result for the medium moving products for the U shape station for various numbers of stations in the raw.

From table 8 we can see that similar number of orders is served by each of the solutions. One station arrangement serves around 0,6 % less orders those two and three
stations in raw solution. The differences are appearing in average total time, which is the smallest for 4 stations in raw arrangement. The reduction of total time is achieved due to dropping picking and waiting times. The average waiting time in case of four stations arrangement is reduced by 99% comparing to one station arrangement. Moreover, single picker utilization is the smallest in case of 4 station arrangement, what suggests that resources are not properly used.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time. The results are depicted in Table 9.

The critical system performances are depicted in figure 5.13. The average number of orders being served by the system is around 25000. In case of total number of units and orders out, the solution with four stations in the raw is twice better than one with two stations in the raw. Hence, looking at the relation of waiting time to the total time, we see that in case of 4 station arrangement the average box spends 89% of the total time in the queues. Respectively, within one station arrangement the box spends 96% of total time in the line. This suggest us high average number of queuing orders. The average number of orders in the queuing are summarize in figure 5.14.

![Figure 5.13. The performance summary.](image-url)
From figure 5.14 and table 9 we see that for 4 station arrangement there is average difference of 0.5 order in number of orders in the queue, while in two stations arrangement the average difference in number of queuing orders is 2.5. If we take a look at the percentage share of the difference between lowest average number in queue and the highest average number in queue in relation to the average lowest number of ordered in queue we find out that under two station in the line this percentage value is around 17%, while under four station arrangement this value is 10%. Moreover, the line when we move closer to the end of the system, the line becomes longer. Fourth picker cumulated queue length is longer than the first one, what stays in contradiction with what we said before: that first picker should be the fastest one – which was supported by the fast moving product zone – figure 5.8. This would lead us to the conclusion that the differences in workload in medium moving zone is smaller than for the fast moving products zone. This suggests lack of the need to put on the first station the fastest picker. We will make detailed analysis of workload in the Section 5.3 – storage policies.

The smaller differences in the amount of work for the four station arrangement are depicted in the average utilization of pickers, depicted figure 5.15. In table 9, under four stations in the raw arrangement the deviation from the average utilization is the smallest one. In the result the pickers has the same amount of the work to be done. Hence, utilization level of 85% suggests that there is still place for the improvements. Improving utilization might be done through exceeding the number of orders in the queues. The differences in single picker utilization are much lower than in case of fast moving products zone depicted in figure 5.10 (page 55).
Figure 5.15. The average utilization of pickers.

The single picker average utilization is lower than in case of fast moving zone products, due to lower amount of work in the medium moving products zone. In the average the order picks 4.8 items in fast moving zone and 0.92 in medium moving zone. The value 0.92 which is divided between four stations give us in average 0.23 item per station. This means that three from four orders do not have any items to pick – the picker picks only the picking list in order to receive the information that there is nothing to be picked from his/her station. Rising up the number of picks per station will lead us to higher picker utilization.

The summary of the results is depicted in table 9.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The U shape</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stations in the raw</strong></td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11.8</td>
<td>4.3</td>
</tr>
<tr>
<td>2</td>
<td>2.7</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td></td>
</tr>
<tr>
<td>4512</td>
<td>12484</td>
</tr>
<tr>
<td>19952</td>
<td>24584</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td></td>
</tr>
<tr>
<td>4.15</td>
<td>11.521</td>
</tr>
<tr>
<td>18358</td>
<td>22667</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td></td>
</tr>
<tr>
<td>284.79</td>
<td>140.92</td>
</tr>
<tr>
<td>112.37</td>
<td>68.91</td>
</tr>
<tr>
<td>Minimum total time</td>
<td></td>
</tr>
<tr>
<td>0.53</td>
<td>1.21</td>
</tr>
<tr>
<td>2.26</td>
<td>3.00</td>
</tr>
<tr>
<td>Maximum total time</td>
<td></td>
</tr>
<tr>
<td>1203.25</td>
<td>548.55</td>
</tr>
<tr>
<td>370.32</td>
<td>252.25</td>
</tr>
<tr>
<td>Average picking time</td>
<td></td>
</tr>
<tr>
<td>11.32</td>
<td>7.89</td>
</tr>
<tr>
<td>7.28</td>
<td>7.45</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td></td>
</tr>
<tr>
<td>0.51</td>
<td>1.05</td>
</tr>
<tr>
<td>1.73</td>
<td>2.37</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td></td>
</tr>
<tr>
<td>101.92</td>
<td>67.14</td>
</tr>
<tr>
<td>51.82</td>
<td>47.25</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td></td>
</tr>
<tr>
<td>273.47</td>
<td>133.03</td>
</tr>
<tr>
<td>105.09</td>
<td>61.45</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td></td>
</tr>
<tr>
<td>1200.56</td>
<td>542.08</td>
</tr>
<tr>
<td>366.56</td>
<td>246.94</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>94.64%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>91.34%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>89.59%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>84.86%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>-</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>-</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>89.79%</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>84.85%</td>
</tr>
<tr>
<td>89.62%</td>
<td>84.87%</td>
</tr>
<tr>
<td>84.89%</td>
<td></td>
</tr>
<tr>
<td><strong>Queue times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td></td>
</tr>
<tr>
<td>285.45</td>
<td>66.12</td>
</tr>
<tr>
<td>34.69</td>
<td>16.55</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>1200.56</td>
<td>430.07</td>
</tr>
<tr>
<td>245.07</td>
<td>164.80</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>76.93</td>
</tr>
<tr>
<td>39.20</td>
<td>17.78</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>523.32</td>
</tr>
<tr>
<td>262.74</td>
<td>181.47</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>40.30</td>
<td>17.83</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>254.21</td>
<td>144.25</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>18.31</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>180.34</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td></td>
</tr>
<tr>
<td>23.01</td>
<td>14.12</td>
</tr>
<tr>
<td>11.64</td>
<td>6.46</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>103.00</td>
<td>104.00</td>
</tr>
<tr>
<td>98.00</td>
<td>86.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>16.71</td>
</tr>
<tr>
<td>13.42</td>
<td>7.08</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>126.00</td>
</tr>
<tr>
<td>98.00</td>
<td>77.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13.82</td>
<td>7.13</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>109.00</td>
<td>78.00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7.31</td>
<td></td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>73.00</td>
<td></td>
</tr>
</tbody>
</table>

*Table 9.* The simulation result for the medium moving products for the U shape station for various numbers of stations in the raw.

From table 9 we can see that the U shape arrangement with four stations in the raw is the best. It allows packing 22600 units per day. The demanded daily quantity is 11000 units daily. There is no need to implement four station arrangements. For our system
suitable solution is two line of single station or one line of two stations in the raw. Since the two stations in the raw offers lower buffer value, better solution is two stations in the raw arrangement.

5.2.6 Slow moving products station layout

On the base of the split in Section 5.1 the company has 1050 slow moving products. The 1050 low demand products are allocated to the slow moving segment. The slow moving segment performs 4 % of total units – from Section 5.1 – 2580 units daily.

The average order size is on the base of assumption in Section 2.1 is described by poisson distribution with the mean value of 6 item. From this value 4 % of item is done by the slow moving products zone. It is equal to 0.24 item per order. The average number of picks within slow moving zone will be described by the poisson distribution with mean value of 0.24 – 4 % from 6, which is mean order size. Since in slow moving products segment there is a bypass each order passing the zone has some picks to be done. In average every fourth order would enter the slow moving zone. The rest of the orders will pass the slow moving zone with the bypass. The bypass will improve single picker utilization.

Considering Caron, F; Marchet, G; Perego, A (2000) there are two main station arrangements for rare picking operations. The first one is static racks organized vertically, the second one is horizontal static rack arrangement. Slow moving products, comparing to products with high sale do not require lot of space for storage which offers the flow rack. One packer may be enough to support two or three days of sale. Hence the amount of slow moving products suggests huge demand for the place. Just for comparison the fast and medium moving products zone contain respectively 140 and 210 products. This makes the static rack the perfect solution for the slow moving products zone. Moreover looking at the reference slow moving station layout it is visible that the slow moving station requires a lot of walking from the picker.

The static rack layout is given in Section 1.9. We maintaining the same dimensions of the bins as in fast and medium moving products zone – 40cm. The only difference is the depth of the bin, which in case of static rack is equal to the depth of the carton. The static rack has four floors of shelves, which is determined by the carton dimension and the average picker high. Each static rack has depth of two bins – it allows for placing two cartons with different product. One of the cartons is served from the front, and the second one is served from the back.

The reference station layout for slow moving products zone is showed in figure 5.16.
Since single station of slow moving products has to contain 1050 products, the size of the station is 8.8 x 6.4 meters. The picker performs a lot of walking within station. On each single static shelf module there is a place for 16 products on each side. Considering picking from both sides we have 32 bins on each static shelf module. There are 6 rows containing 4 modules each. This arrangement will be called further vertical arrangement.

The alternative solution might be horizontal arrangement. Within this arrangement the static shelf modules are arranged horizontally – parallel to the conveyor. The horizontal arrangement for the slow moving products zone is depicted in figure 5.17.
Both arrangements require a lot of space. In each arrangement the implemented storage policy was within aisle policy - according to the reference policy from Section 4.2. In case of vertical arrangement the products with higher sale values were settled in the raw closer to home base of the picker. In the figure 5.16 the products with the highest sale values will be put on the bins of the middle rows. The within aisle is implemented also for the horizontal arrangement – figure 5.17. The higher sale products were put into bins closer to the middle-cross aisle.

**The model architecture**

The details are depicted in figure 5.18. The most of the modules used are exactly the same as for the model from figure 5.2.2. The modules that differ are depicted in the figure 5.18. The description of the modules is as follows:

1. Process module
   Three process modules which are responsible for simulating the road which picker has to pass in order to enter first, second and third aisle. The aisles are shown on figure 5.17.
2. Decide module
   Three decides modules which simulate the choice between different bins locations.
3. Process module
Eight process modules which simulate different availability areas. Each of the static shelves sections form figure 5.17 contains two availability area, for instance three and four steps area.

The rest of the modules used is the same as in model in Subsection 5.2.2.

The simulation results

The simulation was run under the same conditions as for fast and medium moving products zones – the day was 15 working hour long, there were 10 working days. The pickers were simulated considering failure rate. In the reference sorting solution – figure 4.4 is the bypass implemented which transports to the station only boxes which have at least one item to pick. In the slow moving product zone the picker does not have to serve the boxes, which has no items to pick. In the reference model there is only one picker in the raw. The simulation results are depicted in Table 10.
Statistic | The vertical model | The horizontal model
--- | --- | ---
Stations in the raw | 1 | 1
Time between arrivals (Random EXPO) | 14.2 | 14.2

**General**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Orders out</td>
<td>3 756</td>
<td>3 766</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 861</td>
<td>3 871</td>
</tr>
</tbody>
</table>

**Times (seconds)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average total time (left scale)</td>
<td>577.32</td>
<td>327.65</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5.40</td>
<td>5.56</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>2024.58</td>
<td>1603.29</td>
</tr>
<tr>
<td>Average picking time</td>
<td>14.17</td>
<td>13.93</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4.99</td>
<td>4.71</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>58.91</td>
<td>61.10</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>563.15</td>
<td>313.72</td>
</tr>
<tr>
<td>Minimumum waiting time</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>2011.97</td>
<td>1593.21</td>
</tr>
</tbody>
</table>

**Resources**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 utilization</td>
<td>98.58%</td>
<td>97.15%</td>
</tr>
</tbody>
</table>

**Queues times (seconds)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 average queuing time</td>
<td>566.69</td>
<td>319.62</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>2011.97</td>
<td>1593.21</td>
</tr>
</tbody>
</table>

**Number in queues**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Picker 1 average queuing number</td>
<td>39.65</td>
<td>22.15</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>142.00</td>
<td>113.00</td>
</tr>
</tbody>
</table>

*Table 10. The slow moving products zone simulation results.*

Looking at table 10 we can say that two solutions considered are very similar if considered number of orders being served. The reduction is done in case of average total time, which is reduced in case of horizontal arrangement by 43 %. Despite average total time reduction, the picking times are reduced just slightly. The most of the reduction is realize by drop of average waiting times. Moreover, the utilization of single picker states for both solutions at similar level. The difference is around 1.5%. Under simulation with the same between arrival times, the horizontal system due to reduction of average total time is better solution than the vertical one.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time.

The simulation results are highlighted in figure 5.19. The horizontal layout of the station performs better than the vertical one. The horizontal layout serves similar amount of order but average total and waiting time is half, comparing to the vertical model. The order does not wait long time in the line to be served. This influences average line length – average number in the queue. The horizontal model has shorter queues, in the result less space on the conveyor is requires for the queue of boxes. The detailed simulation results are depicted in table 11.
Both arrangements have similar times for the average picking activity. The maximum picking times differ. The horizontal arrangement has almost 30% higher maximum picking time. This might be explained as a result of implemented storage policy – within aisle storage, which is not the most suitable for the horizontal arrangement. According to the Petersen, C.G. (2002) under this conditions better solution would be across aisle storage policy. This issue we will analyze in Section 5.2.

The main difference is in waiting times. The average order spends in horizontal station layout almost half of the time comparing to the vertical arrangement. This is also supported by the queue length, which is doubled for the vertical model. Hence the single picker utilization suggests that the horizontal model is more effective. Within vertical model the picker has to travel more in order to makes the pick. The lower utilization for the horizontal arrangement suggests that there is still place for improvements. In order to improve the picker utilization we might for instance extend the queue size. It will require more space on the conveyor, but it will provide more stable supply in boxes.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The vertical model</th>
<th>The horizontal model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>14,2</td>
<td>14,1</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>3 756</td>
<td>3 788</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 861</td>
<td>3 895</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>577,32</td>
<td>331,22</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5,40</td>
<td>5,76</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>2024,58</td>
<td>1528,36</td>
</tr>
<tr>
<td>Average picking time</td>
<td>14,17</td>
<td>13,91</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4,99</td>
<td>4,80</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>58,91</td>
<td>77,50</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>563,15</td>
<td>317,31</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>2011,97</td>
<td>1517,70</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>98,58%</td>
<td>97,59%</td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>566,69</td>
<td>322,14</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>2011,97</td>
<td>1517,70</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>39,65</td>
<td>22,64</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>142,00</td>
<td>111,00</td>
</tr>
</tbody>
</table>

Table 11. The slow moving products zone simulation results.

The slow moving zone should be able to serve 4% of total demand, which is equal to 2580 units per day. Both vertical and horizontal arrangements are able to pack desired number of units. The horizontal solution of static shelves is slightly better than the vertical arrangement. Hence the average waiting time of horizontal arrangement is almost half of the value than for the vertical solution. This is caused mostly by the shorter walking paths to the places where products are stored. The waiting time influenced the picker utilization, which suggests that there is still place for improvements. That is why the horizontal arrangement is better solution from considered for the slow moving products zone.
5.2.7 Number of slow moving products stations.

We will widen our analysis on the number of station in the raw for the best solution from Subsection 5.2.6. The scenarios examined will include 1, 2, or 3 stations put in the raw. We do not consider 4 stations arrangement due to limitations of academic version of Arena. The simulations conditions are exactly the same as for the simulation from Subsection 5.2.6. – simulation was run for the time equivalent to the 2 shifts work for ten days.

The model architecture

The model architecture is like the one from Subsection 5.2.6. The difference is only in the percentages split of decides modules, which is cause of different station sale values.

The simulation results

We assumed that the stations are situated in the raw, and each station has a bypass. The order will enter only those stations which are necessary. The simulation results are depicted in Table 12.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The horizontal model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td>1</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>14,1</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>3 788</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 895</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>331,22</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5,76</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>1528,36</td>
</tr>
<tr>
<td>Average picking time</td>
<td>13,91</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4,80</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>77,50</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>317,31</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0,00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>1517,70</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>97,59%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>-</td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>322,14</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>1517,70</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>-</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>22,64</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>111,00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>-</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>-</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>-</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 12.* The simulation result for the slow moving products for the horizontal layout station for various numbers of stations in the raw.

From table 12 we can see that similar number of orders is served by each of the solutions. The differences are below 1 %. Looking at average total time, we see that it is the smallest for 3 stations in raw arrangement. The reduction of total time is achieved due to dropping waiting times. Moreover, single picker utilization is the smallest in case of 3 station arrangement, what suggests that resources are not properly used. Looking at small differences between different solutions, in amount of orders being served and at reduction only in waiting times we can conclude that the system performance depend mostly on time between arrivals.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time.
The main system performance is depicted in figure 5.20.

![Graph showing horizontal station main performance](image)

**Figure 5.20. Horizontal station main performance**

The total number of served orders is best for the 3 stations in the raw arrangement. Once again it is visible that the more stations we implement the greater is the number of boxes served. More stations in line make single station size smaller. Moreover, two station arrangement has lower total and waiting times than the three stations arrangement – it stays in contradiction with simulations result in previous sections. This might be explained as a result of small difference between two and three station arrangement and different characteristic of station within slow moving products zone. The differences are so small due to implemented storage policy – within aisle storage policy.

Moreover the utilization for the 3 station arrangement suggests that there is still place for improvements. We can observe that the single picker utilization dependent mostly on number of stations in the raw. It is visible that for three stations in the raw the average utilization is about 88-89 %. We can draw conclusion that the single picker utilization depends partly from the number of the stations in the raw. The utilization summary for horizontal layout is depicted in figure 5.21.
Figure 5.21. The single picker utilization summary.

The summary of the results is depicted in table 13. We see that three station arrangement has the best performance from the tested arrangements. The only draws back are total and waiting times. The long total times are reflected by the queue lengths.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The horizontal model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stations in the raw</strong></td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>14.1  7.1  5.1</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>3 788  7 581  10 464</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 895  7 583  10 469</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>331.22  209.50  218.18</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5.76  1.22  2.23</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>1528.36  860.26  742.13</td>
</tr>
<tr>
<td>Average picking time</td>
<td>13.91  13.01  13.75</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4.80  1.08  1.71</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>77.50  103.71  113.03</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>317.31  196.49  204.43</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00  0.00  0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>1517.70  854.71  733.21</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>97.59%  91.30%  89.65%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>91.52%  89.35%  87.84%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td></td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>322.14  104.39  74.76</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00  0.00  0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>1517.70  620.84  567.25</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>-  106.62  79.01</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>-  0.00  0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>-  648.28  451.42</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>- -  69.39</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>- -  0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>- -  466.76</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>22.64  13.59  13.23</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00  0.00  0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>111.00  91.00  107.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>-  14.09  14.20</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>-  0.00  0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>-  89.00  100.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>- -  12.30</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>- -  0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>- -  90.00</td>
</tr>
</tbody>
</table>

*Table 13.* The simulation result for the slow moving products for the horizontal layout station for various numbers of stations in the raw.

The required number of units to be packed by the slow moving products zone is 2580 units per day. The one station solution is able to serve desired quantity; therefore there is no need to implement plural station arrangement. The good enough solution for the slow moving products zone is the single station, horizontal arrangement depicted in figure 5.17.
5.3 Storage policies

The reference storage policy is within aisle policy – described in Section 4.2. Hwang, H. (2004) considers also across the aisle and perimeter storage policy. Petersen, C.; et. al. (2004) develop the picking system analysis on the diagonal and rectangular storage policies. The examples of above storage policies are showed in figure 5.22.

![Storage policies diagram](image)

Figure 5.22. Storage policies.

In figure 5.22 number one represents the top sale products. Number three respectively low sale products within the zone. The reference storage policy is the within the aisle policy described in Section 4.2 – figure 4.3 (page 33). There is also possibility to implement random policy, but we will not examine it since it usually gives poor results Loon C., Tang and Ek-Peng, Chew, (1997). If station is small enough differences between particular storage policy and random policy might become insignificant. Under this condition the random policy should be considered due to simplicity.

Our analysis considers three picking zones. The fast moving and medium moving zone are built from one raw in front and one raw behind of flow racks. There is only one raw of flow racks, so there is no possibility to implement any other policy than random or within aisle, since in order to implement one of the policies described in Section 5.22 the station need to have at least couple of rows and aisles. Thus we examined storage policies only within the slow moving products zone. Since we do not consider
random policy, the within aisle policy is the only one possible. In case of slow moving products zone there is wider range of storage policies to choose, due to amount of static racks on which the products are stored. All the storage policies from figure 5.22 are possible.

The model architecture

The detailed model architecture is depicted below in figure 5.23.

Figure 5.23. The diagonal station layout model architecture.
The model general architecture is similar to this presented in Subsection 5.2.2. The differences are the modules marked in figure 5.23., and are described as follows:

1. Decide modules
   Three different decide modules simulate three different diagonal storage groups, like in figure 5.22.

2. Process module
   23 process modules simulate three road that picker has to pass in order to reach one of the aisles.

3. Process module
   Eight process modules which simulate different availability area. Each of the static shelves sections from figure 5.17 contains two availability area, for instance three and four steps area.

Simulation results

We run the simulation within the same conditions as for the simulations run in Section 5.2 - 10 working days, 15 effective working hours per day. The simulation was run for the best station layout for the slow moving products zone – the horizontal static rack arrangement. The solution is depicted in Subsection 5.2.6. The simulation results are depicted in Table 14.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The horizontal model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stations in the raw</strong></td>
<td>Within the aisle</td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>14,6</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>3 696</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 805</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>160,08</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5,25</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>714,71</td>
</tr>
<tr>
<td>Average picking time</td>
<td>13,91</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4,77</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>62,56</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>146,17</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0,00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>701,73</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>95,20%</td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>152,14</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>701,73</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>10,08</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>55,00</td>
</tr>
</tbody>
</table>

*Table 14.* The simulation results for different storage polices implemented in the slow moving products zone.
From table 14, we see that all storage policies are able to serve more less the same amount of orders. The difference is lower than 1%. The differences appear in average total times. The diagonal storage policy offers the highest reduction of average total time comparing to within the aisle storage policy – reference storage policy. The reduction is around 50%. The rest of considered storage policies have higher average total times than the reference storage policy. In case of diagonal storage policy most of the time savings is done through reduction of average waiting time. Moreover, single picker utilization is as well the lowest, which suggests that there is still place for improvements – for instance by rising number of orders in the queue. From table 14 the best storage policy is diagonal storage policy.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time.

The highest number of units packed is performed within diagonal storage policy. The slow moving products zone with implemented diagonal storage policy is able to served almost 8% units more than the slow moving products zone with implemented within aisle storage policy – figure 5.24. Hence the total time, which average order spends in the queue is also reduced. The relation of waiting time to the total time across different storage policies seems to be constant, and equal approximately 95%. This would suggest that in order to maintain continuous supply of the orders we must raise length of the conveyor for buffer. The space is required since orders do not appear in equal in between arrival times – random distribution. The space is needed in order to accumulate orders in times when the orders appear more often than average between arrival times. Moreover picker is continuously supplied with the orders from gathered in buffer in times when orders appear rarely.
Figure 5.24. The storage policies main performance.

Figure 5.25 depicts average picker utilization for each storage policy. Within the aisle storage policy represents second from the top single picker utilization. The diagonal storage policy provides lower amount of work for the picker which is highlighted through the lowest single picker utilization. The value 96.5% suggests that there is still small place for improvements. The improvements might be done through expanding the space for buffer and allowing staying within the system more that 150 orders at one time. The details are highlighted in figure 5.25.

Figure 5.25. Pickers utilization.
Figure 5.26. Average number of orders waiting.

The detailed performance of different storage policies are showed in table 15.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>Within the aisle</th>
<th>Across the aisle</th>
<th>Rectangular</th>
<th>Perimeter</th>
<th>Diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>14,1</td>
<td>14,2</td>
<td>14,6</td>
<td>14,4</td>
<td>13,4</td>
</tr>
<tr>
<td>General</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>3 788</td>
<td>3 751</td>
<td>3 643</td>
<td>3 698</td>
<td>3 995</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>3 895</td>
<td>3 853</td>
<td>3 751</td>
<td>3 799</td>
<td>4 106</td>
</tr>
<tr>
<td>Times (seconds)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>331,22</td>
<td>520,64</td>
<td>386,92</td>
<td>347,33</td>
<td>281,88</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>5,76</td>
<td>5,82</td>
<td>5,35</td>
<td>5,97</td>
<td>5,29</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>1528,36</td>
<td>1826,13</td>
<td>1663,04</td>
<td>1555,82</td>
<td>1261,45</td>
</tr>
<tr>
<td>Average picking time</td>
<td>13,91</td>
<td>14,14</td>
<td>14,45</td>
<td>14,22</td>
<td>13,05</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4,80</td>
<td>5,09</td>
<td>4,78</td>
<td>4,74</td>
<td>4,93</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>77,50</td>
<td>62,86</td>
<td>69,67</td>
<td>76,78</td>
<td>58,99</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>317,31</td>
<td>506,50</td>
<td>372,47</td>
<td>333,12</td>
<td>268,83</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>1517,70</td>
<td>1815,88</td>
<td>1641,79</td>
<td>1544,98</td>
<td>1251,86</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>97,59%</td>
<td>98,23%</td>
<td>97,47%</td>
<td>97,36%</td>
<td>96,54%</td>
</tr>
<tr>
<td>Queues times (seconds)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>322,14</td>
<td>510,59</td>
<td>377,94</td>
<td>337,69</td>
<td>276,07</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>1517,70</td>
<td>1815,88</td>
<td>1641,79</td>
<td>1544,98</td>
<td>1251,86</td>
</tr>
<tr>
<td>Number in queues</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>22,64</td>
<td>35,61</td>
<td>25,57</td>
<td>23,13</td>
<td>20,10</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>111,00</td>
<td>129,00</td>
<td>115,00</td>
<td>117,00</td>
<td>92,00</td>
</tr>
</tbody>
</table>

Table 15. The simulation results for different storage polices implemented in the slow moving products zone.
Within the aisle storage policy performance are enough to serve desired number of units. Since all storage policies performance have only slight difference, each of the storage policy might be implemented. Under each storage policy picker will be able to pack desired number of units/orders. The diagonal policy provides lowest number of orders queuing in buffer and the lowest single picker utilization. It means that picker still can be utilized more effectively by raising number of order being stored in buffers. This leads to the conclusion that the best from considered storage policies under our conditions is diagonal storage policy.

The diagonal storage policy might be improved in order to make it even more effective. The improvements might be done through connecting the diagonal storage policy with demand patterns. In the customers orders in reality we should be able to find demand patterns – for instance if the customer ordered product A, there is a 90% probability that he/she will order product B, because there is sale promotion. In order to improve the picking efficiency we should place product B bin as close as it is possible to product A bin. Such solution will improve system utilization as well as reduce the completion time of each order. The picker will not have to walk two times to pick two products. With the storage policy considered demand similarities the picker will have to walk only once in order to pick two items. The algorithm which will choose the items to be placed close to each other will be discussed in Section 5.4 picking policies. In this section the algorithm will be discussed as a tool for batching the orders. The algorithm will be discussed general since we do not have real orders.

5.4 Picking policies

Picking policy answers to the question how orders are grouped into picking tours. According to Petersen, C.G (2000), the picking policy involves assigning items to the picking tours. Petersen, C.G (2000) examines three picking policies: strict order picking, batch picking and zone picking.

Strict order picking assigns a picker only one order during picking tour. The picker picks only one item and returns to the home base location in order to drop the gathered item. The batch picking assigns a picker more than one order during a picking tour. Zone picking assigns picker to a designed picking zone, where the picker is responsible for only those items that are in his/her zone.

The main advantage of strict order picking is that such solution maintains order integrity. It also simplifies the picker’s job and avoids rehandling. Each worker is responsible for an entire order, this allows for simple error checking and establishes responsibility of the order to a single worker. Strict order picking provides fast service when customers are waiting for their orders. However, strict order picking is likely to require a worker to traverse a large portion of the picking station or warehouse to pick an order. Another drawback, when ordered products are varying very much is that it does not allow for speed picking. The picker achieves large productivity when she / he picks in large quantities single item.
Our reference picking policy considers already strict order, in which pickers are divided into zones. Each picker picks only the products from his/her zone. The issue that needs to be considered is the batching problem. Within batching policy, picker picks more than one order during one picking tour. Since our system is divided into zones, the batching policy should be attached at the zone level. The order should be batched in the way that allows picking the biggest number of items within batch. Within the batch there should be at most one product at time picked. Sorting may be done while the picker is traveling within station. The procedure for batch picking is similar to strict order picking. A picker proceeds to every pick location on the pick list. After picking desired quantity of the product A the picker travels to the home base point. During travel the picker splits picked quantity between orders from the batch. When products are placed into the boxes, the picker starts picking next item from the batch of orders. The process proceeds until all products from the batch are picked. Batch picking generally results in less travel time per item, but it means loosing order integrity, and risk for errors is increased. In addition, batch picking with downstream sorting requires place for order consolidation. Effective batch picking requires a balance of travel savings and the cost of sorting and errors. Hence the bulk of success of batching policy depends on implemented batching algorithm. The issue to be considered is the picker capacity – the maximum quantity of item that might be picked and transported at one time.


**Association rule mining algorithm**

Data mining is the procedure for investigating and analyzing a large body of data to discover meaningful patterns and rules. Association rule mining aims to discover some interesting association relationships from large dataset. For order batching, the association rule mining is employed to discover associations between customer’s orders in order database. Therefore, the order-item data table – table 16, is transposed to the item-order data table – table 17.

<table>
<thead>
<tr>
<th>Order</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>A01</td>
<td>A, C, D</td>
</tr>
<tr>
<td>A02</td>
<td>B, C, E</td>
</tr>
<tr>
<td>A03</td>
<td>A, B, C, E</td>
</tr>
<tr>
<td>A04</td>
<td>B, E</td>
</tr>
</tbody>
</table>

*Table 16. An example of order-item data.*
The relationships between customer orders can be found from the item-order data table using the association rule mining. Orders with more same products items could have higher associations, and may form an order batch. The proposed batching approach applies the Apriori algorithm to extract large item sets. The algorithm's aim is: to find all sets of items that have transaction support exceeding the minimum support. The support for an item set then is the percentage of transactions that contain the items in the database. To obtain associations between customer orders for order batching, the association rule mining attempts to find associations for all sets of orders.

Members of the order sets are the orders which contain list of ordered items and quantities. The first stage recognizes the associations between customer orders in terms of support confidence and lift using association rule mining. Meanwhile, the second stage employs a clustering procedure for the grouping orders into batches. The associations between customers orders obtained in the first stage of mining associations rules are used as the basis for the clustering orders. Orders with more similar product items have higher associations. Consequently, batching orders with higher associations can decrease the distance traveled by order picker. The batching algorithm is depicted in figure 5.27.

Table 17. An example of item-order data.

<table>
<thead>
<tr>
<th>Item</th>
<th>Orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A01, A03</td>
</tr>
<tr>
<td>B</td>
<td>A02, A03, A04</td>
</tr>
<tr>
<td>C</td>
<td>A01, A02, A03</td>
</tr>
<tr>
<td>D</td>
<td>A01</td>
</tr>
<tr>
<td>E</td>
<td>A02, A03, A04</td>
</tr>
</tbody>
</table>
Start

Input Order Data set \( A=\emptyset, T=\emptyset, B_k=\emptyset, i=1, j=1 \)

Compute the support values \( s_{ij} \) for order set \( A \) by Apriori algorithm.

Select \( O_{k1}, O_{k2} \), with the highest support value to any order in \( A \), to be the seed of the batch \( B_k \).

Exclude \( O_{k1}, O_{k2} \) from \( A \).

Pick \( O_x \) with the highest support value corresponding to \( B_k \) from \( A-T \), with the highest support value corresponding to \( B_k \).

If \( V(O_x+B_k) \) exceeds pickers capacity, or \( s_{ij}<\theta \)?

Yes: Move \( O_x \) to \( T \), Clear \( O_x \).

No: Move \( O_x \) to \( B_k \), and exclude \( O_x \) from \( A \).

Clear \( T \), clear \( O_x \), \( i=i+1 \)

A-T is null?

Yes: Output \( B_k \), and order which not in each \( B_k, k=1, 2, ..., N \)

No:

\(|A|<1 \) or \( \text{avg}(Z_i)<\theta \)

Yes: Stop

No: Move \( O_x \) to \( B_k \), and exclude \( O_x \) from \( A \).

Figure 5.27. The batching algorithm (ref. Mu-Chen, Ch; et. al. (2005)).
The algorithm notations are listed below:

A  the set of orders that are not assigned to a batch;
T  the set of orders that can not be assigned to the current batch due to capacity restriction;
B_k  the set of orders in the k-th batch;
U  the set of orders finally not contained in any batch;
s_{ij}  support value between customer orders i and j;
V_i  volume of order i;
C_v  capacity of the picker;
K  number of batches under considerations;
N  number of orders;
(O_{k1}, O_{k2}) the pair of seed orders for the k-th batch;
O_x  the candidate order for addition;
V(\bullet) the total volume of orders in the current batch;
Z_{ik}  the sum of the support values of order O_i associated to the k-th batch;
\theta  the threshold for order addition;

The implementation example of batching algorithm is depicted in figure 5.25. The algorithm is done on the base of item-order data table from table 16.

Figure 5.28. The batching algorithm implementation example.

The data mining algorithm uses one attribute, which have to be user specified – support level. Support level is answering for the question how well matched orders we consider in the algorithm procedure. The lower support level, the higher probability that two not exactly the same orders will be picked in the same tour. The above batching algorithm is called Apriori algorithm.
Heuristic batching algorithm

The approach is proposed by Pan, C. and Liu, S. (2000). The authors present a little bit different approach than the data mining algorithm. The procedure is composed of two major steps, the seed selection and the addition of subsequent orders to the batch. The general procedure of a batching heuristic algorithm is depicted in figure 5.29.

![Flowchart](image)

Figure 5.29. An order batching heuristic procedure.

Seed selection rules
Initially, there are no orders in a batch. A seed is defined as the first order to be selected to form the batch. A seed might be chosen according to one of the following four rules:

I. An order is selected as the seed if it has the largest number of items;
II. An order is selected as the seed if its total weight of items is the greatest;
III. An order is considered the seed if it has the largest area of economic convex hull among all orders. The definition and the determination of the Economic Convex Hull algorithm of an order is described by Pan, C. and Liu, S. (2000). The details are described in Appendix 3.
IV. An order is selected as the seed if its 6-D SFC $\theta$ value has the smallest deviation from zero. The definition and the determination of the 6-D SFC $\theta$ is described by Pan, C. and Liu, S. (2000). The details are described in appendix 4.
Order addition rules
After a seed is selected, the remaining orders are included in the batch by one of the following order addition rules:

I. Choose the order which has the largest number of common locations (bins) with the seed;

II. For each candidate order, find the total distance between its item locations and the closest locations in the seed order. The order which has the minimum total distance is added to the current batch;

III. Let $E_j$ denote the economic convex hull of order $j$ and $s$ denote the seed. For every candidate order $i$, calculate the similarity coefficient, $SC$, of order $i$ with seed as

$$SC = \frac{\text{Area of the intersection of } E_s \text{ and } E_i}{\text{Area of the union of } E_s \text{ and } E_i}.$$ 

Choose the order with the largest $SC$ value and add it to the current batch.

IV. Find the 6-D SFC $\theta$ value for every candidate order. The order which has the smallest deviation from the $\theta$ value of the seed is then combined into the batch.

After an order is added to the current batch, the seed is updated such that it includes the newly added order. This is the cumulative seeding rule.

Since we do not have real orders we will assumed that one third all items ordered – order lines is capable to be batched. In average order there will be two products that are going to be batched. There is maximum two orders in the batch. The implications of implementing more that two orders in the batch are higher probability of error and longer sorting process. Moreover the picking process in case of large batches requires more space for storing the boxes which need to be packed. The limitation is station length; we are not able to build the batch which cumulate length of the boxes will be longer than the half of the length of the station. There will be no more space available since the home base of the picker is situated in the middle of the station. Assuming the percentage number of order lines that are batched us is treating to the result of one of the described batching algorithms. Through assuming number of lines being batched we set the result of the batching algorithm. The possibilities of batching the orders depend strongly on order structure and implemented batching algorithm. We do not consider single picker capacity – how many products she / he can pick at one time. The aim of the simulation is to show benefits of order picking under assumed level of order lines that might be batched.

**The model architecture**

The model is similar to this described in Subsection 5.2.3. The differences are covered in figure 5.30. The rest of the model is the same as in model from Subsection 5.2.3.
Figure 5.30. The difference between model from Subsection 5.2.3.

1 Decide module
The module separates orders that are part of the batch. According to our assumptions every third order line might be part of the batch. It means that every fifth passing entity will be treated as a batched order, which is added to the seed order.

2 Process module
The module is responsible for simulating the releasing of box activity. In previous sections this activity was integral part of dropping products into the box activity.

Simulation results

Since we assumed number of order lines being batched it is hard to asses the probability of achieve such a result without having real order set. Within our batch picking we would like to pick at the same tour the orders which have the same order lines. It means that two customers in average ordered exactly the same two products. It is possible within data mining algorithm with setting high support level, but in fact even orders which has only one common product should be batched. The result of the algorithm should be pairs of orders, in which there are the same products ordered. The quantities of ordered products might vary between orders in the pair.

The batching policy gives us couple of savings. The picker picks only one picking list, which is common for pair of the orders in batch. The picker in order to pick product which is batched travels only one time. She/he picks the quantity for two orders at one time. The picker can drop as well the products simultaneously to two orders. The additional process is the sorting needed after the picker traveled back to the home base with already picked product. The sorting process might be done during traveling back – this solution is commonly called by the literature sort while pick. Hence in reality there might be a need to switch orders from batch to batch between stations. It might be especially needed in case of long picking list, when there is no possibility to satisfy
high support level. In order to switch the orders between batched there is a need for additional sorting / transporting solution, which will allow doing this. In case of high support level, and need for maintaining high amount of batches, there will be required loop and the buffer place within each station. For the lower support levels, there will be a need for the loop and the buffer places only between picking zones.

We added batching policy only to the fast moving products zone, since 80% of picking activities is realized in this zone. The fast moving products zone is the most critical and offers the highest possible benefits, when the efficient batching policy is implemented. Our aim is to show the benefits of efficient batching policy implementation. In average picker picks 4.8 item in fast moving products zone. Since the only tool which can batch two orders is the batching algorithm, we should choose one of explained above. Hence, we will not choose one specific algorithm since we do not have set of the orders. The effectiveness of batching policy depends strongly on order structure and length of the picking list. Our aim is to show benefits of general batching policy within the considered order fulfillment system, with assumption that we are able to batch 33% of order lines. Moreover if we would like to add the batching policy to all zones, there will be a need to switch the batched orders between zones. It means that within fast moving products zone the two orders – A and B are batched in batch k-th, and in the medium moving products zone, the orders A is in k+1-th batch and the order B is in k+2-th batch. In practice the batching policy which will allow changes between segments, will require some sortation-buffer area, in which orders might change sequence and be buffered for a while. Hence system which will allow switching the batches between zones will have better performance, and it will require more sophisticated batching algorithm.

In simulation, we implemented exactly the same conditions as for previous sections.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The U shape 4 stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>5.5</td>
</tr>
<tr>
<td>Without batching</td>
<td>5.5</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>9787</td>
</tr>
<tr>
<td>With batching</td>
<td>9851</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>55372</td>
</tr>
<tr>
<td>With batching</td>
<td>55736</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>185.91</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>7.56</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>642.39</td>
</tr>
<tr>
<td>Average picking time</td>
<td>20.31</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4.25</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>81.40</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>165.60</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>625.16</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>91.02%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>91.73%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>93.10%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>92.64%</td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>76.93%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>77.72%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>78.87%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>78.58%</td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>46.28</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>433.73</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>38.40</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>254.94</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>49.75</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>314.44</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td>46.59</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td>346.34</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>88.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>6.30</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>48.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>8.31</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>61.00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td>66.00</td>
</tr>
</tbody>
</table>

Table 18. The simulation results for fast moving products zone with implemented batching policy.

From table 18 we see that system with implemented batching policy offers around 0.5% more orders served comparing to the system without implemented batching policy. The average total time is reduced by 60% comparing to referenced system. The 15% reduction of time is done in core picking activity. The average waiting time is reduced
by 70%. Average waiting time is the result of number of orders in the line, which is as well reduced. Moreover single picker utilization drops as well in case of system with implemented batching policy. Single picker utilization in system with implemented batching policy is around 77%, which suggests that resources are not fully utilized. Considering both systems, which performances are depicted in table 18, the better system is the one with implemented batching policy.

We simulate once again each of the models trying to find the lowest time between arrivals. We check the maximum system performance under Arena academic version limitations. The simulation does not allow simulating more than 150 orders at one time. The main system performance is depicted in figure 5.31.

![Figure 5.31. The main performance of fast moving products line with implemented batching policy.](image)

In the fast moving zone products the batching policy offers 19% improvements in amount of served orders. The improvement is done through the reduction of traveling within station. Within assumed data we achieve as well total and waiting times reduction. The reduction is done by respectively 6 and 5%. The total time reduction should influence the total number of orders waiting in the queue. The line length dropped in average by 38% - the details are depicted in figure 5.32. This means significant reduction in place necessary for buffers. The system with batching policy requires less space on the conveyor for storing boxes.
Figure 5.32. The picker’s in fast moving products zone utilization comparison.

From figure 5.32 the single picker utilization in the system with implemented batching policy is also reduced. The reduction is done by around 1%. The higher single picker utilization might be expanded by rising up the number of orders in the each station buffer. The detailed data summarizing the implementation of batching policy are depicted below in table 19.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>The U shape 4 stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stations in the raw</td>
<td></td>
</tr>
<tr>
<td>Time between arrivals (Random EXPO)</td>
<td>Without batching</td>
</tr>
<tr>
<td></td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>With batching</td>
</tr>
<tr>
<td></td>
<td>4.6</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>9787</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>55372</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>185.91</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>7.56</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>642.39</td>
</tr>
<tr>
<td>Average picking time</td>
<td>20.31</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>4.25</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>81.40</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>165.60</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0.00</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>625.16</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 utilization</td>
<td>91.02%</td>
</tr>
<tr>
<td>Picker 2 utilization</td>
<td>91.73%</td>
</tr>
<tr>
<td>Picker 3 utilization</td>
<td>93.10%</td>
</tr>
<tr>
<td>Picker 4 utilization</td>
<td>92.64%</td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing time</td>
<td>46.28</td>
</tr>
<tr>
<td>Picker 1 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing time</td>
<td>433.73</td>
</tr>
<tr>
<td>Picker 2 average queuing time</td>
<td>38.40</td>
</tr>
<tr>
<td>Picker 2 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing time</td>
<td>254.94</td>
</tr>
<tr>
<td>Picker 3 average queuing time</td>
<td>49.75</td>
</tr>
<tr>
<td>Picker 3 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing time</td>
<td>314.44</td>
</tr>
<tr>
<td>Picker 4 average queuing time</td>
<td>46.59</td>
</tr>
<tr>
<td>Picker 4 minimum queuing time</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing time</td>
<td>346.34</td>
</tr>
<tr>
<td><strong>Number in queues</strong></td>
<td></td>
</tr>
<tr>
<td>Picker 1 average queuing number</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 1 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 1 maximum queuing number</td>
<td>88.00</td>
</tr>
<tr>
<td>Picker 2 average queuing number</td>
<td>6.30</td>
</tr>
<tr>
<td>Picker 2 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 2 maximum queuing number</td>
<td>48.00</td>
</tr>
<tr>
<td>Picker 3 average queuing number</td>
<td>8.31</td>
</tr>
<tr>
<td>Picker 3 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 3 maximum queuing number</td>
<td>61.00</td>
</tr>
<tr>
<td>Picker 4 average queuing number</td>
<td>7.71</td>
</tr>
<tr>
<td>Picker 4 minimum queuing number</td>
<td>0.00</td>
</tr>
<tr>
<td>Picker 4 maximum queuing number</td>
<td>66.00</td>
</tr>
</tbody>
</table>

Table 19. The simulation results for fast moving products zone with implemented batching policy.

The effective batching algorithm seems to be a profitable solution. The algorithm raises the number of orders and units served by the system by around 19%. The main system performance like waiting time and average number of orders waiting are also improved.
5.5 Replenishment process

The replenishment process is necessary in order to provide continuous supply of items to be picked on stations. The replenishment process occurs in parallel with the picking process and in a nearby area. In order to pack into the boxes desired number of products we need one, four stations in the raw fast moving zone products line (with implemented batching policy); one, two stations in the raw medium moving products line, and one, single station slow moving products zone station. We assume that average carton contains 80 pieces of product. In fast moving products zone there is 140 bins to be replenished. There are as well 210 and 1050 bins to be replenished in medium moving and slow moving zone products respectively. As we had defined in scope definitions the flow rack is deep enough to contain four cartons with products. The static shelves might contain only single carton.

According to the demand data from Appendix 1 and taking into consideration average number of products in the carton, the fast moving zone requires 1390 cartons to be replenished daily. The medium moving products zone needs to be supplied with 272 cartons daily. Respectively slow moving products zone needs 64 cartons daily. If we consider daily effective working hours – 15 effective working hours, the packaging line under consideration requires, considering all products, 115 cartons to be replenished per hour.

Byung-In, Kim, et. al. (2003) in his paper developed new replenishment process logic for the efficient warehouse operation. The proposed replenishment and storage allocation logic can be generalized and applied to similar warehousing or distribution environments. The author divides analyzed picking system into 16 picking zones and corresponding to them 16 replenishment zones. The author introduced the idea of pan-swapping. Pan-swapping is the process of switching “empty” flow racks with refilled one. The author treats the replenishment as a process of refilling empty flow racks with full one. The pan-swapping activities will appear in system under consideration, at the end of each month. Then the company sale profile changes. Byung-In, Kim, et. al. (2003) introduces the idea of replenishment zones. We are going to develop this idea. The author uses mostly the replenishment zones in pan-swapping activity; we are going to use it in current replenishment process, since the amount of products needed is much greater, and a few refilling activities are required.

The replenishment zone is the area where the products will be stored in collective cartons. The replenishment zone has to be big enough to store all products in appropriate quantities. It might be seen as a drop buffers between two production stages implemented in just in time systems.

The replenishment process will be two kinds here. One is the replenishment of the flow racks directly from the drop buffer area, and the second is the replenishment of mentioned drop buffers area from “undefined” central warehouse. If we consider the simulation from previous sections in connection with the average number of pieces in carton we will have the knowledge about required frequency of replenishment. In the
result it will allow us to specify the amount of staff needed. We do not analyze this process due to its simplicity. Other replenishment aspect will require specifying replenishment policy – the quantity of products being stored in buffers and the amount in which the products are ordered from central warehouse – the size of the batches. We focus our analysis on the second aspect of replenishment process. On the base of followed assumed lead times, we will determine number of workers required.

We start our analysis from determining the replenishment system logic. We use the replenishment approach proposed by Byung-In, Kim, et. al. (2003) – replenishment zones. Hence in order to make the system even more excellent we will add the kanban philosophy described by Chun-Che, Huang (1996). We treat the order picking system as a manufacturing line, and the replenishment zones as internal buffers of the system, where the products between next production phases are stored. We assume that products are shipped from central warehouse or in some exceptions directly from production line.

We will design kanban system according to Chun-Che, Huang (1996) paper overview. To the kanban logic we will determine values of reorder point (ROP) and economical order quantity (EOQ). We will summarize main functions of kanbans, which in fact are the reason for implementation. The functions of kanbans are as follows:

- **Visibility function** – the information and material flow are combined together as kanbans move with their parts (work-in-process),
- **Production function** – the kanban detached from the succeeding stage fulfills a production control function which indicates the time, quantity, and the part types to be produced.
- **Inventory function** – the number of kanbans actually measures the amount of inventory. Hence, controlling the number of kanbans is equivalent to controlling the amount of inventory; i.e. increasing (decreasing) the number of kanbans corresponds to increasing (decreasing) the amount of inventory. Controlling the number of kanbans is much simpler than controlling the amount of inventory itself.

The inventory function of kanban cards is the most important in case of order picking system under consideration. This function we are going to develop. The aim of kanban implementation in order picking system is to reduce level of inventory and smooth the inventory flow. The kanban cards we are going to use to manage the inventory flow between central warehouse and replenishment zones. We will propose the solution which includes the kanban system for the replenishment of the order picking system.

In manufacturing environment the kanban system is used in production stage i, when parts are processed and demand from its receiving stage i+1 occurs. The production kanban is removed from a container and is placed on the dispatching board at stage i. The withdrawal kanban from stage i+1 then replaces the production kanban and the container. This container along with the withdrawal kanban is then sent to stage i+1 for processing. Meanwhile at stage i, the production activity takes place when a production kanban and a container with the withdrawal kanban are available. The
withdrawal kanban is then replaced by the production kanban and sent back to stage i-1 to initiate production activity at stage i-1. This forms a cyclic production chain. The kanban pulls (withdrawals) parts instead of pushing parts from one stage to another to meet the demand at each stage. In a situation when no withdrawal is requested by the succeeding stage, the preceding stage will not produce at all, and hence no excess items are manufactured. Therefore, with kanban system under just-in-time philosophy, non stock production may be achieved. The kanban system might be implemented for the internal processes (between two stages of production) and external one as well (suppliers).

According to the review of Chun-Che, Huang (1996) there are two possible kanban systems that might be considered for our system: the single kanban system and the semi-dual kanban system.

**The single kanban system**

The single kanban system is used mostly to block material-handling based on the part type. In system under consideration it will mean blocking the product-handling. The product should be blocked between the replenishment zones and the central warehouse based on total queue size. Under single kanban system multiple containers contain the batches to be replenished and picked. The sizes of the containers for different products may vary.

On the base of Chun-Che, Huang (1996) the following conditions are essential for proper functioning of the single kanban system:

- Small distance between the central warehouse (or production drop area) and the replenishment zones,
- Fast turnover of kanabans,
- Low WIP,
- Small buffer space and fast turnover of WIP,
- Synchronization between the order picking rate and speed of material handling,

**The semi-dual kanban system**

For this system the following conditions are essential for proper functioning:

- Large distance between the central warehouse (or production drop area) and the replenishment zones,
- Slow turnover of kanabans,
- Large WIP between the central warehouse (or production drop area) and the replenishment zones,
- Slow turnover of WIP,
- Synchronization between the order picking rate and speed of material handling is not necessary,

We develop for order picking system under consideration single kanaban system. We use the single kanban since in order picking system we have fast turnover of kanbans and fast turnover of WIP. Kanban cards turnover fast, because we have assumed short
lead times values – values less than hour. We use single kanban system despite large WIP, which suggest that we should use semi-dual system. The size of line is established to be 1400 products, and theoretically we can have 1400 working in progress in replenishment. The distance between the central warehouse and the replenishment zones is assumed to be rather small – the replenishment zones and the warehouse are placed in one building. Considering size of line (number of products in the order picking system) 1400 items, we have to cope with large WIP between the central warehouse (or production drop area) and the replenishment zones. Moreover under kanban system connected with reorder point and economical order quantity, the kanban is automatically put in the line. Since the time of needed shipment in the moment of need creation (crossing reorder point) is equal for each of the products, we have in the system under consideration classical first-in-first-out rule.

The general replenishment process under kanban system is as follows:

- The carton with the products is put to the bins by the bin filler directly from the replenishment zones (we set the replenishment localization further).
- The picker packs the item to the box. The box is released from station. The box has completed picking within the fast moving products zone, and it is transported to next picking zone. The system receives information that order has completed picking within fast moving products zone.
- The order fulfillment system takes off the packed quantity from the warehouse balance.
- The system checks if desired buffer size minus actual warehouse quantity is higher than the container size. If “yes” the information need to replenish for this product is send to the central warehousing system. If “no” the system does not sent any information.
- When the central warehousing system receives the kanban card, it should check the desired shipping time and place kanban in the queue. At the beginning of the queue should be placed the kanban with the nearest needed shipping time. The system should monitor online the needed shipping time, which depends on the amount of the products that was packed in time between now and when kanban card flow into the central warehouse system.
- The product is shipped from the central warehouse and put into appropriate replenishment zone.
- The system receives the information that kanban card was fulfilled.

The above process might be done in traditional way, including kanban boxes and the kanban cards as sheets of paper. The above process is designed with consideration of automated warehousing system. If we would like to implement traditional kanban system, it would be acting as in figure 5.33.
Figure 5.33. The single kanban system.

The number of kanban cards is equal to sum of reorder point (ROP) economical order quantity (EOQ) divided by the average number of products in container. The kanaban card in our analysis is equal to the carton.

Number of kanbans = \(\frac{\text{ROP} + \text{EOQ}}{\text{A}}\)

where:

\(\text{A}\) - number of items in a container. We had assumed 80 pieces in average carton.

The reorder point according to Aghazadeh, Seyed-Mamoud (2001) is calculated as follows:

\(\text{ROP} = \text{D} \times \text{L}\)

where:

\(\text{D}\) - it is maximum product quantity being sold daily. In order to make kanban system works, the sale quantity can not vary too much. It is equal to maximum daily number of units being sold. In our analysis we will consider December’s values of sale, since they are maximal.

\(\text{L}\) - lead time. We consider lead times as follows: shipping waiting time, order picking processing time, withdraw lead time.
The lead time is calculated as follows:

\[ L = \alpha + \text{Shipping waiting time} + \text{Order picking processing time} + \text{Withdraw lead time} \]

where:

- **Shipping waiting time** - it is the idle interval between the time when the warehouse received the command to ship item and the time when the product is actually shipped to the replenishment zone. Since the central warehouse is situated in the same building, we assume that for average shipment there is 0.5 hour required.

- **Order picking processing time** - it is the idle interval between the time when picker put the item into the box, and the time when order picking system has information that the product was packed. We consider in our analysis the average time spent in the system which is equal: 175, 69 and 282 seconds, respectively for fast, medium, and slow moving products zone.

- **Withdraw lead time** - is time required to passing the kanban card to the warehouse. If we implement the kanban system supported with the warehousing computer systems, this time is minimal, and equal almost 0.

- **\( \alpha \)** - it is safety stock, which is included in lead time. In our system we will implement the safety factor value 1 hour.

On the base of *Aghazadeh, Seyed-Mamoud (2001)* economical order quantity is calculated as follows:

\[ \text{EOQ} = \sqrt{\left( \frac{D \times S}{H} \right)} \]

where:

- **D** - demand rate.
- **S** - shipping order cost. Shipping cost corresponds to the cost of hiring the driver of the forklift plus the cost of forklift purchase divide by the average productivity of the worker. The forklift cost according to the [http://www.jungheinrich.com.pl/index_PL_PL.html](http://www.jungheinrich.com.pl/index_PL_PL.html) Company is 200000 PLN. The depreciation is assumed to be ten years. The average productivity of the forklift is assumed to be 15 picks per hour. The salary of forklift driver is assumed to be 2800 PLN/month. According to above data the cost per pick is equal to 0.93 PLN. The pick means
transportation of the product from the warehouse to the replenishment zone, depend less of transported quantity. Only the movement fact does matter.

\[ H \]

- holding cost. Holding cost corresponds to expenses taken for building warehouse. We assume that the cost of building one square meter of warehouse is equal to 4680 PLN. In case of one ear depreciation we have daily cost of one location in two floor warehouse 6,4 PLN/day. On one storage location we can keep 40 cartons and in average carton we have as assumed before 80 pieces, so the daily holding cost is equal to 0,002 PLN/piece of product.

We estimate for each product different number of kanban cards, since each zone has different lead times and different number of units sold. Each product has different reorder point and different economical order quantity. The total number of kanbans in system under consideration is 3337. In fast moving products zone we have 1187 kanbans, in medium moving products zone – 859 kanbans, and in slow moving zone 1291. Each kanban means one carton of products. By setting up number of kanbans we determine the space required for storing the products, since each kanban card means one carton. We need 920 storage locations in replenishment zone. We divide products into three classes, and assign them three types of store places. 52 products require more than 9 cartons to be stored at one time are stored in location, which have dimensions equal to euro pallet – 100cm x 100cm x 160cm (breath x depth x height). 202 products require greater number of kanbans than 5, and less than 9 are stored in locations which have dimensions: 50cm x 100cm x 160cm. 666 products are stored in locations which have dimension 25cm x 100cm x 160cm.

The system logic including kanban approach, reorder point and economical order quantity is depicted in figure 5.34. The algorithm should start acting when packed order going out of particular segment.
\( j_i \) number of lines packed in particular zone in the picking list in order i-th.

\( k \) product index. \( k = 1, 2, \ldots, 1400 \).

\( Z_k \) number of stored products type k-th in replenishment zone.

\( O_{ki} \) number of products k purchased in order i-th.

\( ROP_k \) reorder point for item part k-th

\( EOQ_k \) economical order quantity for item part k-th

Figure 5.34. The replenishment system logic.

Queues of orders which are created consider automatically different shipping dates, due to reorder points that differ for different products. Each product has the same chance for being stock out. By changing safety factor lead time, we can adjust the risk of stock outs. The stock out is likely to appear when couple of orders will be placed in the warehouse system at one time. We examine in simulation the replenishment solution and determine number of staff required.
The model architecture

The replenishment model has been created by measuring single station simulation time when there was a need for replenishment the storage locations. The values were recorded from the previous models, taking into considerations reorder points and economical order quantity. The statistics from record were imported into input analyzer in order to find the statistic pattern which describes system behavior. The replenishment of slow moving zone due to big amount of products were simplified by taking average time between replenishment needs – the total number of units packed within the slow moving products zone was (2580) was divided by the average carton size (80). By this value the total simulation time was divided. The statistic data of each module are depicted in table 20. Replenishment process time requirements were covered by the process module according to the assumed lead time from previous section. Detailed replenishment model architecture is depicted in figure 5.35. Detailed model documentation might be found in Appendix 8.

Figure 5.35. The replenishment model architecture.
<table>
<thead>
<tr>
<th>Name</th>
<th>Entity Type</th>
<th>Type</th>
<th>Value</th>
<th>Expression</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 3</td>
<td>Entity 3</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>411 + 221 * BETA(0.156, 0.17)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A1 2</td>
<td>Entity 2</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>150 + 624 * BETA(0.706, 0.856)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A1 1</td>
<td>Entity 1</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>151 + 617 * BETA(1.15, 1.22)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A2 3</td>
<td>Entity 6</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>394 + 225 * BETA(0.137, 0.162)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A2 2</td>
<td>Entity 5</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>149 + 624 * BETA(0.708, 0.936)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A2 1</td>
<td>Entity 4</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>154 + 617 * BETA(1.14, 1.23)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A3 3</td>
<td>Entity 9</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>UNIF(385, 763)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A3 2</td>
<td>Entity 8</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>147 + 626 * BETA(0.805, 0.848)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A3 1</td>
<td>Entity 7</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>154 + 621 * BETA(1.13, 1.21)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A4 2</td>
<td>Entity 11</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>143 + 609 * BETA(0.91, 0.872)</td>
<td>Seconds</td>
</tr>
<tr>
<td>A4 1</td>
<td>Entity 10</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>154 + 610 * BETA(1.12, 1.17)</td>
<td>Seconds</td>
</tr>
<tr>
<td>Slow zone</td>
<td>Entity 10</td>
<td>Random (Expo)</td>
<td>1674.41</td>
<td>154 + 610 * BETA(1.12, 1.17)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B1 3</td>
<td>Entity 3</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>200 + 557 * BETA(0.595, 0.778)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B1 2</td>
<td>Entity 2</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>158 + 608 * BETA(0.754, 0.899)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B1 1</td>
<td>Entity 1</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>143 + 627 * BETA(1.24, 1.16)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B2 3</td>
<td>Entity 6</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>202 + 568 * BETA(0.567, 0.734)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B2 2</td>
<td>Entity 5</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>150 + 595 * BETA(0.723, 0.781)</td>
<td>Seconds</td>
</tr>
<tr>
<td>B2 1</td>
<td>Entity 4</td>
<td>Expression</td>
<td>-10 + LOGN(66500, 1740000)</td>
<td>146 + 619 * BETA(1.31, 1.22)</td>
<td>Seconds</td>
</tr>
</tbody>
</table>

Table 20. The create module statistic data.

Each of the above create modules simulates replenishment demand from one of the particular stations. Additionally create modules have been divide into sections in order to simulate more accurately replenishment demand from particular group of the products.

Simulation results

The simulation was done for the same conditions as in the previous sections – two shifts, 15 effective working hours. Between each replication, which corresponds to day the system was not cleared. It means that stocks were passing between days. The state from the end of the day before means the beginning of day after. This kind of simulation is the closest to reality.

For the analysis we use the Process analyzer included in Arena package. The software allows monitor different simulated system performance under changing conditions. In our analysis we need to determine the number of workers needed to transport the products from the warehouse to the replenishment zones. Under designed system, considering assumed data we have to determine the number of workers necessary to serve the system. The simulation highlights are depicted in table 21.
<table>
<thead>
<tr>
<th>Name</th>
<th>Program file</th>
<th>Replications</th>
<th>Number of resources</th>
<th>Average queue length</th>
<th>Average resource utilization</th>
<th>Max number in queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Replenishment adjustment 2 : sam bin file</td>
<td>1 68.0000</td>
<td>NA</td>
<td>1</td>
<td>8.420</td>
<td>0.992</td>
<td>24,0</td>
</tr>
<tr>
<td>2 Replenishment adjustment 2 : sam bin file</td>
<td>1 69.0000</td>
<td>NA</td>
<td>1</td>
<td>2.173</td>
<td>0.971</td>
<td>19,0</td>
</tr>
<tr>
<td>3 Replenishment adjustment 2 : sam bin file</td>
<td>1 70.0000</td>
<td>3.946</td>
<td>1</td>
<td>3.946</td>
<td>0.981</td>
<td>14,0</td>
</tr>
<tr>
<td>4 Replenishment adjustment 2 : sam bin file</td>
<td>1 71.0000</td>
<td>0.867</td>
<td>1</td>
<td>0.867</td>
<td>0.953</td>
<td>19,0</td>
</tr>
<tr>
<td>5 Replenishment adjustment 2 : sam bin file</td>
<td>1 72.0000</td>
<td>0.566</td>
<td>1</td>
<td>0.566</td>
<td>0.940</td>
<td>11,0</td>
</tr>
<tr>
<td>6 Replenishment adjustment 2 : sam bin file</td>
<td>1 73.0000</td>
<td>0.368</td>
<td>1</td>
<td>0.368</td>
<td>0.931</td>
<td>11,0</td>
</tr>
<tr>
<td>7 Replenishment adjustment 2 : sam bin file</td>
<td>1 74.0000</td>
<td>0.217</td>
<td>1</td>
<td>0.217</td>
<td>0.916</td>
<td>9,0</td>
</tr>
<tr>
<td>8 Replenishment adjustment 2 : sam bin file</td>
<td>1 75.0000</td>
<td>0.124</td>
<td>1</td>
<td>0.124</td>
<td>0.907</td>
<td>12,0</td>
</tr>
<tr>
<td>9 Replenishment adjustment 2 : sam bin file</td>
<td>1 76.0000</td>
<td>0.067</td>
<td>1</td>
<td>0.067</td>
<td>0.893</td>
<td>7,0</td>
</tr>
<tr>
<td>10 Replenishment adjustment 2 : sam bin file</td>
<td>1 77.0000</td>
<td>0.045</td>
<td>1</td>
<td>0.045</td>
<td>0.882</td>
<td>6,0</td>
</tr>
<tr>
<td>11 Replenishment adjustment 2 : sam bin file</td>
<td>1 78.0000</td>
<td>0.014</td>
<td>1</td>
<td>0.014</td>
<td>0.870</td>
<td>6,0</td>
</tr>
<tr>
<td>12 Replenishment adjustment 2 : sam bin file</td>
<td>1 79.0000</td>
<td>0.003</td>
<td>1</td>
<td>0.003</td>
<td>0.860</td>
<td>7,0</td>
</tr>
<tr>
<td>13 Replenishment adjustment 2 : sam bin file</td>
<td>1 80.0000</td>
<td>0.005</td>
<td>1</td>
<td>0.005</td>
<td>0.847</td>
<td>2,0</td>
</tr>
<tr>
<td>14 Replenishment adjustment 2 : sam bin file</td>
<td>1 81.0000</td>
<td>0.001</td>
<td>1</td>
<td>0.001</td>
<td>0.841</td>
<td>4,0</td>
</tr>
<tr>
<td>15 Replenishment adjustment 2 : sam bin file</td>
<td>1 82.0000</td>
<td>0.003</td>
<td>1</td>
<td>0.003</td>
<td>0.826</td>
<td>4,0</td>
</tr>
<tr>
<td>16 Replenishment adjustment 2 : sam bin file</td>
<td>1 83.0000</td>
<td>0.000</td>
<td>1</td>
<td>0.000</td>
<td>0.805</td>
<td>1,0</td>
</tr>
</tbody>
</table>

Table 21. The process analyzer results

According to the process analyzer data 69 resources is able to serve the system. We are not able to simulate less than 69 resources due to limitations of academic Arena versions limitations, which not allow simulating more than 150 entities at one time in the system. If we have 69 resources, the average number of needed picks in queue is less than 8. The average picking time is around half hour on one entity of resource. We assumed already another hour as a safety stock. In situation when we have 69 resources, one hour of safety stock is enough to pick queuing products. Hence, the bin filler utilization suggests that the resources are properly used – single resource utilization 99%. The maximum number in queue under this condition is 24. Considering the safety lead time – 1 hour, in which we can transport 138 picks, we are able to transport the goods, and due to implemented safety lead time we avoid stock out situation.

Even with academic version of Arena we can notice that proposed system still have safety buffer, mostly due to implemented high lead time. There might be implemented two solutions: we can reduce the lead time and as a result inventory level or we can reduce number of resources required. The conditions should be examined under full version of Arena.
5.6 Sorting solution

The goal of sorting solution is to transport the boxes between the zones and stations. The transportation should provide equal zones utilization. Sorting solution should be able to handle temporary picks in workload.

The reference model corresponds to the solution suggested by Chin-Chia, Jane (2003), the box with the order should visit one by one each of the zones. Author does not specify the sequence of the visited zones, but each order passes the same sequence of stations. We are going to examine reference sorting system in which order is forced to pass fast moving products zone and medium moving product zone as in the reference sorting solution. In the slow moving product zone we will add the by pass which will allow orders that does not have any product from that segment to pass the zone. Orders in the system will be forced to pass sequentially each of the zones. The order can not pass to medium moving products zone until all the products are picked from fast moving products zone. Despite lack of products to pick in fast or medium moving products zone the order will have to go through it. The same rule will be applied on the station level – each station will have to be visited one by one.

The suggested model we are going to develop according to Gang Jing, Gary (1998) guidelines. The author suggests the idea of one main line conveyor, from which the boxes can enter desired zone. Our model performance will be compared with the reference solution. We will expand Gang Jing, Gary (1998) system by adding additional main loop, which should allow orders to choose the sequence of visited zones or stations freely. The order will be allowed for instance to start picking from the slow moving products zone. Hence we will add in fast and medium moving zone, the logic which will allow changing the sequence of stations visited. This should provide even workload distribution. This system ability will be especially valuable during pick periods.

The model architecture

The model contains the list of processes, assign and decide modules. The process modules represent the stations. Each process simulates different station. We have 4 stations in fast moving products zone, two in medium moving products zone and one station in slow moving products zone. The data for each of the process were collected by recording the statistics of previous models. Collected data were analyzed with input analyzer, which in the result gave us probability function. The probability function describes the tame that is needed for particular station to serve the order. After each of the station – process there is assign module which assigns to the entity variable with defined value. This value confirms that this order collected already the products from particular station. Then entity pass the sequence of decide modules which refer it to next station. Decides modules aim is to choose for the entity next station which has the lowest number of orders waiting in the queue. Similar sequence of decide modules is placed just after create module. These modules are choosing the zone to be visit first.
Decide modules are choosing the one which cumulated queues value is the lowest. The detailed model documentation is depicted in Appendix 9.

**Simulation results**

The simulation was done for 15 effective working hours. There were four replications, which is equal to four days. Each station was simulated independently. There were four stations in fast moving products zone, two in medium moving products zone and one in slow moving zone. Each of the station had separate and independent queue. The orders were assigned to the station on the base of present line length. The order was assigned to the station which has the shortest queue length. This mechanism provides equal workload between stations.

The detailed assignment mechanism should be looking as follows:

1. Import order into the system,
2. Choose the minimum cumulated zone queue length among zones that order suppose to visit. The values that we are going to compare in case of order that suppose to visit all the zones, will be: sum of queue length of four stations in fast moving zone against sum of queue length of two stations in medium moving zone against sum of queue length of one stations in slow moving zone. The order should be transported to the zone with the lowest cumulated queue length.
3. Choose the minimum station queue within the zone. The station considered are only those that the order has to go thought in order to complete picking within the zone. The order should be assign to the station which has at the moment minimum queue length.
4. When the order completed the picking within the first station, the system should assign it to the next station, which has at this time minimum queue length. The procedure should be applied until all the stations within particular zone have been visited.
5. When all the stations within the zone have been visited, the system should assign the order to the next zone applying rules from point 2. The procedure should be applied until all the zones have been visited.

The figure 5.36 depicts average picker utilization for reference and suggested sorting solution. The proposed sorting solution, which considers even workload algorithm, provides little bit better workload of the pickers. The differences are rather small. The greater differences are visible in figure 5.37, which summarize average orders line length. It is visible that the proposed sorting solution offers great reduction in case of line lengths. It is especially visible for the fast moving products zone.
The proposed sorting solution offers up to 50% of the queue length reduction. Hence, reduction is also achieved in the medium and slow moving products zone. Within these zones reduction in line length is up to 13%. The queue reduction for medium and slow moving products zone is lower because the utilization within these zones is also very low. High line reduction in fast moving products zone is possible due to high utilization of the pickers. Moreover, higher than the average number of queuing orders to the first station in the fast moving zone, cause of station selection rule implemented. When all the station is idle, the queue length to each station is zero, the order will be assigned to first station. Hence higher than the average number of orders in the A1 station queue is also caused by the statistical disturbances. The A1 station is the one, which is likely to be first station for order that enters fast moving products zone. The A1 is influenced the most by the statistical disturbances, produced by the order start point. This, automatically overstate the average value. This drawback of the algorithm in connection with statistical disturbances provides higher than the average number of queuing orders in station A1.

Figure 5.36. Average picker utilization.
The detailed simulation results of the order picking system are depicted in Table 22.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>Reference sorting solution</th>
<th>Proposed sorting solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orders out</td>
<td>9,254</td>
<td>9,268</td>
</tr>
<tr>
<td>Units out (right scale)</td>
<td>55,521</td>
<td>55,609</td>
</tr>
<tr>
<td><strong>Times (seconds)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average total time (left scale)</td>
<td>259,3</td>
<td>237,9</td>
</tr>
<tr>
<td>Minimum total time</td>
<td>15,5</td>
<td>14,3</td>
</tr>
<tr>
<td>Maximum total time</td>
<td>757,7</td>
<td>2,597,3</td>
</tr>
<tr>
<td>Average picking time</td>
<td>33,6</td>
<td>33,9</td>
</tr>
<tr>
<td>Minimum picking time</td>
<td>5,2</td>
<td>5,5</td>
</tr>
<tr>
<td>Maximum picking time</td>
<td>107,3</td>
<td>194,8</td>
</tr>
<tr>
<td>Average waiting time (left scale)</td>
<td>225,7</td>
<td>204,0</td>
</tr>
<tr>
<td>Minimum waiting time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>Maximum waiting time</td>
<td>723,4</td>
<td>2,488,6</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker A1 utilization</td>
<td>95,1%</td>
<td>98,8%</td>
</tr>
<tr>
<td>Picker A2 utilization</td>
<td>95,4%</td>
<td>95,1%</td>
</tr>
<tr>
<td>Picker A3 utilization</td>
<td>94,7%</td>
<td>95,3%</td>
</tr>
<tr>
<td>Picker A4 utilization</td>
<td>91,9%</td>
<td>92,5%</td>
</tr>
<tr>
<td>Picker B1 utilization</td>
<td>71,9%</td>
<td>72,3%</td>
</tr>
<tr>
<td>Picker B2 utilization</td>
<td>72,3%</td>
<td>72,5%</td>
</tr>
<tr>
<td>Picker C1 utilization</td>
<td>55,7%</td>
<td>55,7%</td>
</tr>
<tr>
<td><strong>Queues times (seconds)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picker A1 average queuing time</td>
<td>76,2</td>
<td>71,5</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>462,5</td>
<td>390,5</td>
</tr>
<tr>
<td>Picker A2 average queuing time</td>
<td>55,8</td>
<td>38,7</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>387,2</td>
<td>240,7</td>
</tr>
<tr>
<td>Picker A3 average queuing time</td>
<td>43,2</td>
<td>42,2</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>300,2</td>
<td>255,6</td>
</tr>
<tr>
<td>Picker A4 average queuing time</td>
<td>24,1</td>
<td>26,6</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>176,0</td>
<td>190,6</td>
</tr>
<tr>
<td>Picker B1 average queuing time</td>
<td>10,4</td>
<td>9,6</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>146,2</td>
<td>126,2</td>
</tr>
<tr>
<td>Picker B2 average queuing time</td>
<td>14,2</td>
<td>10,8</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>144,2</td>
<td>176,3</td>
</tr>
<tr>
<td>Picker C1 average queuing time</td>
<td>10,1</td>
<td>10,4</td>
</tr>
<tr>
<td>minimum queuing time</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>maximum queuing time</td>
<td>140,1</td>
<td>142,4</td>
</tr>
<tr>
<td>Picker</td>
<td>average queuing time</td>
<td>minimum queuing time</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Picker A1</td>
<td>13,1</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker A2</td>
<td>9,6</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker A3</td>
<td>7,4</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker A4</td>
<td>4,1</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker B1</td>
<td>1,7</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker B2</td>
<td>2,4</td>
<td>0,0</td>
</tr>
<tr>
<td>Picker C1</td>
<td>0,4</td>
<td>0,0</td>
</tr>
</tbody>
</table>

Table 22. The sorting solution simulation results.

If we consider data from Table 22, we see that proposed system performances are better than the referenced arrangement system. The critical performance – number of orders being served is only slightly smaller in case of proposed sorting solution, but still the system is able to serve the amount of orders required in pick month. The utilization values stays more less at the same level in case of both systems. The average total time is reduced by the 20%, which in the result will mean lower value of WIP. The total time reduction is achieved by lowering the waiting times. The average picking times stay unchanged. The average times and number of queuing orders give us the information about the space on the conveyor that will be required. The space required is different for every station. The proposed sorting solution is not free of drawbacks but it offers significant benefits – average within time reduction, and in the result WIP reduction.

The system under consideration final layout is depicted in figure 5.38.
Figure 5.38. Final layout of the order picking system.
6 Conclusion

This paper compared the performance of different solution in station layout, storage policies, picking process, replenishment process and sorting solution. First, based on literature the station layouts were compared in order to develop the best performance one. We found that the U-shape station layout offers 10 % better productivity comparing to tunnel layout, due to reduction of traveling distances. Then different storage policies were analyzed in order to choose one with the best performance. For the slow moving products zone we found that the diagonal storage policy provides the greatest reduction of time for picking. Then through comparison of different picking policies, the benefits of batching algorithm were shown. In replenishment section the collective container transportation process was considered. In the analysis of replenishment section we have examined implementation possibilities of pull system based on Kanban philosophy. The entire model contained the layout of the sorting conveyor. In addition benefits of the even workload distribution were shown.

There are several practical limitations of this study. This study does not take into consideration congestion between the aisles during replenishment process, results from having more than one worker replenishing. Another limitation was ignoring the added time associated with height of the storage locations and the distance from one side of the aisle to another. Hence, this study did not analyze the traveling policies. In order to analyze the problem the traversal, midpoint and optimal routing strategies should be taken into consideration.
7 Further research

Further research should be done to determine influence of particular batching algorithm on picking performance. In our work we have assumed that 30% of the lines might be batched. The batching possibility depends strongly on order structures. In order to determine influence of batch algorithm will be required the sample of the orders. The influence of the length of the picking list should be also analyzed.

Moreover further research should be done in order to find out the model behavior under extended simulation conditions. The above simulation approach is constrained by the limitations of academic version of Arena. The order picking system should be simulated without the limitations of Academic version of Arena. The model in which all the modules are together should be examined in order to have complete knowledge about system behavior.

The further research should be done in order to measure influence of different techniques of passing information: picking list, pick to voice, pick to screen, pick to light.

Hence, further research should be done in order to establish likelihood of errors under each of the solution. The errors have not been taken into consideration since in order picking systems they mostly depend on human factor.
REFERENCES


Che-Hung Lin and Iuan-Yuan Lu, (1999); “The procedure of determining the order picking strategies in distribution center”; *International Journal Production Economics* 60-61, 301-307

Chin-Chia Jane, Yih-Wenn Laih. Jan2003. „A clustering algorithm for item assignment in a synchronized zone order picking system“; *O.R. Applications*


Geinzer Christine M., Meszaros John P.. 1990. „Modeling high volume conveyor sorting systems“, *Proceedings of the 22nd conference on Winter simulation*; p 714-719

123


“Howard dimension atlas” (2001); Centralny Instytut Ochrony Pracy, Warsaw.


Masel Dale, Goldsmith David. “Using a simulation model to evaluate the configuration of a sortation facility”; Proceeding of the 1997 Winter Simulation Conference

Meller, R.D. Apr1997. “Optimal order-to-lane assignments in an order accumulation/sortation system”; IIE transactions; Vol. 29, n 4, p 293-301


Petersen, C.G. 2000 “Considerations in order picking zone configuration “; International Journal of Operations & Production Management; v 22, n 7, p 793-805


“Talked voices better picks”. Feb2004. Food Manufacture; Vol. 79 Issue 2, p25, 1/5p
## Appendix 1. Monthly demand data

<table>
<thead>
<tr>
<th>Year</th>
<th>January</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>August</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2001</td>
<td>5550</td>
<td>6105</td>
<td>5950</td>
<td>5895</td>
<td>5845</td>
<td>5795</td>
<td>5745</td>
<td>5695</td>
<td>5645</td>
<td>5595</td>
<td>5545</td>
<td>5495</td>
</tr>
<tr>
<td>2002</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2003</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004</td>
<td>46</td>
<td>233</td>
<td>1686</td>
<td>141</td>
<td>294</td>
<td>1962</td>
<td>5230</td>
<td>570</td>
<td>17344</td>
<td>290</td>
<td>307</td>
<td>23073</td>
</tr>
<tr>
<td>2005</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2006</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2007</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2008</td>
<td>10</td>
<td>101</td>
<td>5641</td>
<td>19</td>
<td>38</td>
<td>553</td>
<td>16058</td>
<td>76</td>
<td>2034</td>
<td>5210</td>
<td>12257</td>
<td>5196</td>
</tr>
<tr>
<td>2009</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2012</td>
<td>155</td>
<td>17</td>
<td>64</td>
<td>243</td>
<td>1042</td>
<td>7067</td>
<td>2055</td>
<td>243</td>
<td>8533</td>
<td>3128</td>
<td>5308</td>
<td>155</td>
</tr>
<tr>
<td>2013</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2014</td>
<td>11</td>
<td>101</td>
<td>5641</td>
<td>19</td>
<td>38</td>
<td>553</td>
<td>16058</td>
<td>76</td>
<td>2034</td>
<td>5210</td>
<td>12257</td>
<td>5196</td>
</tr>
<tr>
<td>2015</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2016</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2017</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2018</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2019</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2020</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2021</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2022</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2023</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2024</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2025</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2026</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2027</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2028</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2029</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2030</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:** The table above represents the monthly demand data for the years 2000 to 2030. Each entry corresponds to the demand for a specific month, with columns indicating the months and rows indicating the years.
<table>
<thead>
<tr>
<th></th>
<th>33</th>
<th>73</th>
<th>113</th>
<th>154</th>
<th>195</th>
<th>236</th>
<th>277</th>
<th>318</th>
<th>359</th>
<th>399</th>
<th>440</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Appendix 2. Demand skewness

Figure A1. Demand skewness in third quarter.
Appendix 3. The economic convex hull algorithm

Consider the picker as an AS/RS (Automatic storage-retrieval machine). Let \( m = \tan^{-1}(V_x/V_y) \) and line L be the straight line passing through the home base point in the station with slope m. Let \( P_1, P_2 \) and \( P_3 \) denote the boundary of an order where \( P_1 \) lies on line 1, \( P_2 \) lies on line 2 and \( P_3 \) lies on line 3 in such a way that the line 1 and line 2 are both parallel to L, and line 3 has the slope of \(-m\), as shown in figure 15.1. The region bounded by the horizontal axis, the vertical axis, line 1, line 2 and line 3 is called the economic convex hull \( E_i \) of an order i.

![diagram](image.png)

*Figure A2. The economic convex hull of an order.*

The area of the economic convex hull of each order is calculated and the third seed selection rule selects the one which has the largest area of economic convex hull of all the orders as the seed. Let \( s \) denote the seed and \( E_{s|i} \) represent the intersection of \( E_s \) and \( E_i \) of an order i. The similarity measure SC is defined as:

\[
SC = \frac{\text{Area} \(E_{s|i}\)}{\text{Area} \(E_s\) + \text{Area} \(E_i\) - \text{Area} \(E_{s|i}\)}
\]

SC is computed for every order remaining to be batched. The order which has the largest SC value is selected as the one to be added to the current batch by the third order addition rule.
Appendix 4. The 6-D SFC $\theta$ algorithm

In accordance with Pan, C. and Liu, S. (2000), the space filling curve represents a continuous mapping $\psi$ from points on a unit circle onto the points on a unit $d$-dimensional cube. A point $\theta$ on the unit circle is measured $\theta$ revolution clockwise removed, where $0 \leq \theta \leq 1$, from the origin. An example of space filling is illustrated in figure 15.2.

The space filling curve has the property that the inverse image of $\psi$ can be computed easily. The 6-D SFC batching heuristic applies the inverse 6 dimensional space filling curve to transform the six numbers $(x_{\text{min}}, y_{\text{min}}, x_{\text{mean value}}, y_{\text{mean value}}, x_{\text{max}}, y_{\text{max}})$ of an

Figure A3. Inverse space filling curve mapping.

The space filling curve has the property that the inverse image of $\psi$ can be computed easily. The 6-D SFC batching heuristic applies the inverse 6 dimensional space filling curve to transform the six numbers $(x_{\text{min}}, y_{\text{min}}, x_{\text{mean value}}, y_{\text{mean value}}, x_{\text{max}}, y_{\text{max}})$ of an
order to a single $\theta$ value on the unit circle, where these six values are the minimum, the arithmetic means and the maximum x and y coordinates, respectively, of all the item locations in that order.

The fourth seed selection rule picks the order whose $\theta$ value is closest to the origin ($\theta=0$) as the seed of the batch. The fourth order addition rule then finds the order which has minimum distance in terms of $\theta$ values to any orders in the current batch among all candidate orders and includes it in the batch.
Appendix 5. Fast moving products zone, U shape, 3 stations
in the raw arrangement documentation (Siman language)

The description of the modules is given with blue font.

; Model statements for module: Create 1
CREATE module creates the entity. The entity is created with between arrival time 7.2
seconds, with exponential probability distribution.

52$ CREATE, 1:SecondstoBaseTime(0.0),Entity
1:SecondstoBaseTime(EXPO(7.2)):NEXT(53$);

53$ ASSIGN: Start.NumberOut=Start.NumberOut + 1:NEXT(11$);

; ; Model statements for module: Assign 20
ASSIGN module assigns to variable 2 new value. When entity is passing through
assign module, to variable 2 is added value of 1. The variable simulates the length of
the line before station.

; 11$ ASSIGN: Variable 2=variable 2+1:NEXT(7$);

; ; Model statements for module: Decide 3
DECIDE module checks weather the picker is busy or idle. If the picker 1 is idle order
is transported to the picker, other case it is stopped in the queue in HOLD module.

; 7$ BRANCH, 1:
    If,Variable 1==1,56$,.Yes:
    Else,57$,Yes;
56$ ASSIGN: If picker1 busy?.NumberOut True=If picker1
busy?.NumberOut True + 1:NEXT(4$);

57$ ASSIGN: If picker1 busy?.NumberOut False=If picker1
busy?.NumberOut False + 1:NEXT(46$);

; ; Model statements for module: Hold 1
HOLD modules stores entities before station 1 until signal with decelerated value appears from SIGNAL module. The HOLD module releases one entity per one signal.

; 4$ QUEUE, Hold 1.Queue;
WAIT: 1,1:NEXT(46$);

;  ; Model statements for module: Assign 31
ASSIGN module assigns four variables to the entity. Variable Licznik 1 is responsible for counting how many items have been picked already. Variable Repeat 1 is responsible for determining number of items to be picked within the station. Variable 2 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 1 determines the state of picker (busy / idle).

; 46$ ASSIGN: Licznik 1=0:
  Repeat 1=anint(pois(1.6)):
  Variable 2=variable 2-1:
  Variable 1=1:NEXT(3$);

;  ; Model statements for module: Process 10
PROCESS module simulates the activity of picking the picking list by picker.

; 3$ ASSIGN: Picking a picking list.NumberIn=Picking a picking list.NumberIn + 1:
  Picking a picking list.WIP=Picking a picking list.WIP+1;
61$ QUEUE, Picking a picking list.Queue;
60$ SEIZE, 3,VA:
  Resource 1,1:NEXT(59$);
59$ DELAY: Triangular(0.5,1,1.5),,VA;
58$ RELEASE: Resource 1,1;
106$ ASSIGN: Picking a picking list.NumberOut=Picking a picking list.NumberOut + 1:
  Picking a picking list.WIP=Picking a picking list.WIP-1:NEXT(0$);

;  ; Model statements for module: Decide 1
DECIDE module determines from which group of the bins the order has product to be picked.
Model statements for module: SIGNAL 1

SIGNAL module is responsible for sending the signal with value 1 to HOLD module.

6$

SIGNAL: 1:NEXT(20$);

Model statements for module: Assign 23

ASSIGN module assigns to variable 4 new value. When entity is passing through assign module, to variable 4 is added value of 1. The variable simulates the length of the line before station.

20$

ASSIGN: Variable 1=0:
Variable 4=variable 4+1:NEXT(17$);

Model statements for module: Decide 5

DECIDE module checks weather the picker is busy or idle. If the picker 2 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.

17$

BRANCH, 1:
If, Variable 3==1, 111$, Yes:
Else, 112$, Yes;

111$

ASSIGN: Decide 5.NumberOut True=Decide 5.NumberOut True + 1:NEXT(14$);

112$

ASSIGN: Decide 5.NumberOut False=Decide 5.NumberOut False + 1:NEXT(38$);

Model statements for module: Hold 2
HOLD modules stores entities before station 2 until signal with decelerated value appears from SIGNAL module. The HOLD module release one entity per one signal.

14$ QUEUE, Hold 2.Queue;
   WAIT: 2,1:NEXT(38$);

38$ ASSIGN: Variable 4=variable 4-1:
          Variable 3=1:
          Licznik 2=0:
          Repeat 2=anint(pois(1.6)):NEXT(44$);

44$ TALLY: FSC na stacje 2,Repeat 2,1:NEXT(13$);

13$ ASSIGN: Picking lista2.NumberIn=Picking lista2.NumberIn + 1:
          Picking lista2.WIP=Picking lista2.WIP+1;

116$ QUEUE, Picking lista2.Queue;
115$ SEIZE, 3,VA:
       Resource 2,1:NEXT(114$);

114$ DELAY: Triangular(0.5,1.1.5),,VA;
113$ RELEASE: Resource 2,1;
161$ ASSIGN: Picking lista2.NumberOut=Picking lista2.NumberOut + 1:
          Picking lista2.WIP=Picking lista2.WIP-1:NEXT(12$);
DECIDE module determines from which group of the bins the order has product to be picked.

12$ \text{BRANCH, 1:}
   \text{With, 66.54/100, 22$, Yes:}
   \text{With, 19.6/100, 18$, Yes:}
   \text{With, 9.86/100, 19$, Yes:}
   \text{Else, 16$, Yes;}

SIGNAL module sends the signal, when order pass through the module to hold module;
16$ \text{SIGNAL: 2:NEXT(32$);}

ASSIGN module assigns to variable 6 new value. When entity is passing through assign module, to variable 6 is added value of 1. The variable simulates the length of the line before station.
32$ \text{ASSIGN: Variable 3=0:}
   \text{Variable 6=variable 6+1:NEXT(28$);}

DECIDE module checks weather the picker is busy or idle. If the picker 3 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.
28$ \text{BRANCH, 1:}
   \text{If, Variable 5==1, 166$, Yes:}
   \text{Else, 167$, Yes;}
166$ \text{ASSIGN: Decide 7.NumberOut True=Decide 7.NumberOut True + 1:NEXT(25$);}
167$ \text{ASSIGN: Decide 7.NumberOut False=Decide 7.NumberOut False + 1:NEXT(39$);}
HOLD modules stores entities before station 3 until signal with decelerated value appears from SIGNAL module. The HOLD module release one entity per one signal.

\[ 25\$ \text{ QUEUE, Hold 3.Queue; } \]
\[ \text{WAIT: 3,1:NEXT(39\$); } \]

ASSIGN module assigns four variables to the entity. Variable Licznik 3 is responsible for counting how many items have been picked already. Variable Repeat 3 is responsible for determining number of items to be picked within the station. Variable 6 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 5 determines the state of picker (busy / idle).

\[ 39\$ \text{ ASSIGN: Variable 6=variable 6-1: } \]
\[ \text{Variable 5=1: } \]
\[ \text{Licznik 3=0: } \]
\[ \text{Repeat 3=anint(pois(1.6)):NEXT(45\$); } \]

RECORD module records data about number of product being picked in station 3 within each order.

\[ 45\$ \text{ TALLY: FSC na stacje 3,Repeat 3,1:NEXT(24\$); } \]

PROCESS module simulates the activity of picking the picking list by the picker on station 3.

\[ 24\$ \text{ ASSIGN: Picking lista3.NumberIn=Picking lista3.NumberIn + 1: } \]
\[ \text{Picking lista3.WIP=Picking lista3.WIP+1; } \]
\[ 171\$ \text{ QUEUE, Picking lista3.Queue; } \]
\[ 170\$ \text{ SEIZE, 3,VA: } \]
Resource 3,1:NEXT(169$);

169$ DELAY: Triangular(0.5,1,1.5),,VA;
168$ RELEASE: Resource 3,1;
216$ ASSIGN: Picking lista3.NumberOut=Picking lista3.NumberOut + 1:
Picking lista3.WIP=Picking lista3.WIP-1:NEXT(23$);

; ;
; Model statements for module: Decide 6
DECIDE module determines from which group of the bins the order has product to be
picked.
;
23$ BRANCH, 1:
    With,64.91/100,34$,Yes:
    With,20.67/100,30$,Yes:
    With,10.43/100,31$,Yes:
Else,27$,Yes;

; ;
; Model statements for module: Signal 3
SIGNAL module sends the signal to hold module.
;
27$ SIGNAL: 3:NEXT(29$);

; ;
; Model statements for module: Assign 25
ASSIGN module assigns to the entity variable with value 0, which determines state of
the picker (idle)
;
29$ ASSIGN: Variable 5=0:NEXT(2$);

; ;
; Model statements for module: Dispose 1
DISPOSE module – the end of the system
;
2$ ASSIGN: Dispose 1.NumberOut=Dispose 1.NumberOut + 1;
221$ DISPOSE: Yes;
Model statements for module: Process 60
PROCESS module simulates activity of picking product placed within hand area.

ASSIGN: Process 60.NumberIn = Process 60.NumberIn + 1:
Process 60.WIP = Process 60.WIP + 1;
QUEUE, Process 60.Queue;
SEIZE, 2, VA:
Resource 3,1: NEXT(223$);
DELAY: Triangular(0,0.2,0.4), VA;
RELEASE: Resource 3,1;
ASSIGN: Process 60.NumberOut = Process 60.NumberOut + 1:
Process 60.WIP = Process 60.WIP - 1: NEXT(33$);

Model statements for module: Process 59
PROCESS module simulates activity of picking product placed within one step area.

ASSIGN: biny 03.NumberIn = biny 03.NumberIn + 1:
biny 03.WIP = biny 03.WIP + 1;
QUEUE, biny 03.Queue;
SEIZE, 2, VA:
Resource 3,1: NEXT(274$);
DELAY: Triangular(1,1.4,2), VA;
RELEASE: Resource 3,1;
ASSIGN: biny 03.NumberOut = biny 03.NumberOut + 1:
biny 03.WIP = biny 03.WIP - 1: NEXT(51$);

Model statements for module: Record 6
RECORD module records data about number of boxes that pass station 3 without picking any product.

COUNT: Pominiete3,1: NEXT(42$);

Model statements for module: Assign 30
ASSIGN module assigns to licznik 3 new value. When entity is passing through assign module, to licznik 3 is added value of 1. The variable simulates the number of products already picked within the station.

; 42$ ASSIGN: Licznik 3=Licznik 3+1:NEXT(43$);

; 43$ BRANCH, 1:
    If,Licznik 3<Repeat 3,324$,Yes:
    Else,325$,Yes;
324$ ASSIGN: Powtorzenie 3.NumberOut True=Powtorzenie 3.NumberOut True + 1:NEXT(24$);
325$ ASSIGN: Powtorzenie 3.NumberOut False=Powtorzenie 3.NumberOut False + 1:NEXT(27$);

; Model statements for module: Decide 9
DECIDE module checks whether all products from particular station have been picked.
; 30$ ASSIGN: Process 52.NumberIn=Process 52.NumberIn + 1:
    Process 52.WIP=Process 52.WIP+1;
329$ QUEUE, Process 52.Queue;
328$ SEIZE, 2,VA:
    Resource 3,1:NEXT(327$);
327$ DELAY: Triangular(1.4,2,2.8),,VA;
326$ RELEASE: Resource 3,1;
374$ ASSIGN: Process 52.NumberOut=Process 52.NumberOut + 1:
    Process 52.WIP=Process 52.WIP-1:NEXT(37$);

; Model statements for module: Process 52
PROCESS module simulates activity of picking product placed within two steps area.
; 37$ ASSIGN: Process 63.NumberIn=Process 63.NumberIn + 1:
Process 63.WIP=Process 63.WIP+1;
380$ QUEUE, Process 63.Queue;
379$ SEIZE, 2,VA:
    Resource 3,1:NEXT(378$);
378$ DELAY: Triangular(1,2,3),,VA;
377$ RELEASE: Resource 3,1;
425$ ASSIGN: Process 63.NumberOut=Process 63.NumberOut + 1:
    Process 63.WIP=Process 63.WIP-1:NEXT(33$);

; ; Model statements for module: Process 53
PROCESS module simulates activity of picking product placed within four steps area.
;
31$ ASSIGN: Process 53.NumberIn=Process 53.NumberIn + 1:
    Process 53.WIP=Process 53.WIP+1;
431$ QUEUE, Process 53.Queue;
430$ SEIZE, 2,VA:
    Resource 3,1:NEXT(429$);
429$ DELAY: Triangular(3,4,4.8),,VA;
428$ RELEASE: Resource 3,1;
476$ ASSIGN: Process 53.NumberOut=Process 53.NumberOut + 1:
    Process 53.WIP=Process 53.WIP-1:NEXT(37$);

; ; Model statements for module: Process 43
PROCESS module simulates activity of picking product placed within hands area.
;
22$ ASSIGN: Process 43.NumberIn=Process 43.NumberIn + 1:
    Process 43.WIP=Process 43.WIP+1;
482$ QUEUE, Process 43.Queue;
481$ SEIZE, 2,VA:
    Resource 2,1:NEXT(480$);
480$ DELAY: Triangular(0,0.2,0.4),,VA;
479$ RELEASE: Resource 2,1;
527$ ASSIGN: Process 43.NumberOut=Process 43.NumberOut + 1:
    Process 43.WIP=Process 43.WIP-1:NEXT(21$);
PROCESS module simulates activity of picking product placed within one step area.

ASSIGN: biny 02.NumberIn=biny 02.NumberIn + 1:
biny 02.WIP=biny 02.WIP+1;

QUEUE, biny 02.Queue;
SEIZE, 2,VA:
Resource 2,1:NEXT(531$);

DELAY: Triangular(1,1.4,2),,VA;
RELEASE: Resource 2,1;
ASSIGN: biny 02.NumberOut=biny 02.NumberOut + 1:
biny 02.WIP=biny 02.WIP-1:NEXT(50$);

RECORD module records data about number of boxes that passing station 2 without picking any product.
COUNT: Pominiete2,1:NEXT(40$);

ASSIGN module assigns to licznik 2 new value. When entity is passing through assign module, to licznik 2 is added value of 1. The variable simulates the number of products already picked within the station.
ASSIGN: Licznik 2=Licznik 2+1:NEXT(41$);

DECIDE module checks whether all products from particular station have been already picked
BRANCH, 1:
If,Licznik 2<Repeat 2,581$,Yes:
Else,582$,Yes;
ASSIGN: Powtorzenie 2.NumberOut True=Powtorzenie 2.NumberOut True + 1:NEXT(13$);
ASSIGN: Powtorzenie 2.NumberOut False=Powtorzenie 2.NumberOut False + 1:NEXT(16$);

; ; Model statements for module: Process 35
PROCESS module simulates activity of picking product placed within two steps area.
; ASSIGN: Process 35.NumberIn=Process 35.NumberIn + 1: Process 35.WIP=Process 35.WIP+1;
QUEUE, Process 35.Queue;
SEIZE, 2,VA: Resource 2,1:NEXT(584$);
DELAY: Triangular(1.4,2,2.8),,VA;
RELEASE: Resource 2,1;

; ; Model statements for module: Process 62
PROCESS module simulates activity of turning around.
QUEUE, Process 62.Queue;
SEIZE, 2,VA: Resource 2,1:NEXT(635$);
DELAY: Triangular(1,2,3),,VA;
RELEASE: Resource 2,1;

; ; Model statements for module: Process 36
PROCESS module simulates activity of picking product placed within four steps area.
688$ QUEUE, Process 36.Queue;
687$ SEIZE, 2,VA:
    Resource 2,1:NEXT(686$);

686$ DELAY: Triangular(3,4,4.8),,VA;
685$ RELEASE: Resource 2,1;
733$ ASSIGN: Process 36.NumberOut=Process 36.NumberOut + 1:
    Process 36.WIP=Process 36.WIP-1:NEXT(36$);

; ; Model statements for module: Process 19
PROCESS module simulates activity of picking product placed within hands area.
;
8$ ASSIGN: Hands area.NumberIn=Hands area.NumberIn + 1:
    Hands area.WIP=Hands area.WIP+1;
739$ QUEUE, Hands area.Queue;
738$ SEIZE, 2,VA:
    Resource 1,1:NEXT(737$);

737$ DELAY: Triangular(0,0.2,0.4),,VA;
736$ RELEASE: Resource 1,1;
784$ ASSIGN: Hands area.NumberOut=Hands area.NumberOut + 1:
    Hands area.WIP=Hands area.WIP-1:NEXT(1$);

; ; Model statements for module: Process 2
PROCESS module simulates activity of dropping products into box.
;
1$ ASSIGN: Dropping item into box.NumberIn=Dropping item into box.NumberIn + 1:
    Dropping item into box.WIP=Dropping item into box.WIP+1;
790$ QUEUE, Dropping item into box.Queue;
789$ SEIZE, 2,VA:
    Resource 1,1:NEXT(788$);

788$ DELAY: Triangular(1,1.4,2),,VA;
787$ RELEASE: Resource 1,1;
835$ ASSIGN: Dropping item into box.NumberOut=Dropping item into box.NumberOut + 1:
    Dropping item into box.WIP=Dropping item into box.WIP-1:NEXT(49$);
RECORD module records data about number of product being picked in station 1 within each order.

COUNT: Record1,1:NEXT(47$);

ASSIGN module assigns to licznik 1 new value. When entity is passing through assign module, to licznik 1 is added value of 1. The variable simulates the number of products already picked within the station.

ASSIGN: Licznik 1=Licznik 1+1:NEXT(48$);

DECIDE module checks whether all products from particular station have been picked.

BRANCH, 1:
If,Licznik 1<Repeat 1,838$,Yes:
Else,839$,Yes;

ASSIGN: Number of items from the station.NumberOut True=Number of items from the station.NumberOut True + 1 :NEXT(3$);

ASSIGN: Number of items from the station.NumberOut False= Number of items from the station.NumberOut False + 1:NEXT(6$);

PROCESS module simulates the activity of picking the product from the one step area.

ASSIGN: One step area.NumberIn=One step area.NumberIn + 1:
One step area.WIP=One step area.WIP+1;

QUEUE, One step area.Queue;
process module simulates the activity of turning around by the picker.

; Model statements for module: Process 21
PROCESS module simulates the activity of picking the product from two steps area.

; Model statements for module: Process 61
PROCESS module simulates the activity of turning around by the picker.
VARIABLES:  Process 53.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Number of items from the station.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 62.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  Process 36.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Picking a picking list.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Process 60.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  Process 63.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): If picker1 busy?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):  
  Two step area.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Dropping item into box.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
    Dropping item into
box.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 43.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
    biny 02.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Powtorzenie 2.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):  
  Picking lista3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Decide 7.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 52.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Hands area.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Decide 7.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): Powtorzenie 3.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):  
  biny 03.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Powtorzenie 3.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):  
  Picking lista3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 53.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 35.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Process 43.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Dropping item into
box.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Turning around.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Process 62.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
Hands area.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Process 52.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
Process 35.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Dispose 1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
Process 63.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Process 36.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
Hands area.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
Picking a picking list.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
biny 03.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Picking lista3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Repeat 1,CLEAR(System),CATEGORY("User Specified-User Specified"): Repeat 2,CLEAR(System),CATEGORY("User Specified-User Specified"): Repeat 3,CLEAR(System),CATEGORY("User Specified-User Specified"): Picking lista2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Process 63.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Decide 5.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): Process 43.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Picking a picking list.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): One step area.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Process 53.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Process 52.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): biny 03.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): biny 02.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Turning around.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Two step area.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Start.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Decide 5.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): Two step area.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Picking lista2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Number of items from the station.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): Process 60.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): One step area.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): If picker1 busy?.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): biny 02.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Turning around.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Process 35.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Picking lista2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): Process 60.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Process 36.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Process 62.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): Variable 1,CLEAR(System),CATEGORY("User Specified-User Specified"): Variable 2,CLEAR(System),CATEGORY("User Specified-User Specified"): Variable 3,CLEAR(System),CATEGORY("User Specified-User Specified"): Variable 4,CLEAR(System),CATEGORY("User Specified-User Specified"): Variable 5,CLEAR(System),CATEGORY("User Specified-User Specified"): Variable 6,CLEAR(System),CATEGORY("User Specified-User Specified"): One step area.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Licznik 1,CLEAR(System),CATEGORY("User Specified-User Specified"): Licznik 2,CLEAR(System),CATEGORY("User Specified-User Specified"): Licznik 3,CLEAR(System),CATEGORY("User Specified-User Specified"):
Powtorzenie 2.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude");


PICTURES: Picture.Airplane:
Picture.Green Ball:
Picture.Blue Page:
Picture.Telephone:
Picture.Blue Ball:
Picture.Yellow Page:
Picture.EMail:
Picture.Yellow Ball:
Picture.Bike:
Picture.Report:
Picture.Van:
Picture.Widgets:
Picture.Envelope:
FAILURES: Failure 1, Time(EXPO(3000),EXPO(20),);
           Failure 2, Time(EXPO(3000),EXPO(20),);
           Failure 3, Time(EXPO(3000),EXPO(20),);

RESOURCES: Resource
           1,Capacity(1),,,COST(0.0,0.0,0.0,0.0),CATEGORY(Resources), FAILURE(Failure
           1,Preempt),AUTOSTATS(Yes,);
           Resource
           2,Capacity(1),,,COST(0.0,0.0,0.0,0.0),CATEGORY(Resources), FAILURE(Failure
           2,Preempt),AUTOSTATS(Yes,);
           Resource
           3,Capacity(1),,,COST(0.0,0.0,0.0,0.0),CATEGORY(Resources), FAILURE(Failure
           3,Preempt),AUTOSTATS(Yes,);

COUNTERS: Record1,,,,DATABASE(,"Count","User Specified","Record1");
          Pominiete2,,,,DATABASE(,"Count","User Specified","Pominiete2");
          Pominiete3,,,,DATABASE(,"Count","User Specified","Pominiete3");

TALLIES: FSC na stacji 2,,DATABASE(,"Expression","User Specified","FSC na
          stacji 2");
          FSC na stacji 3,,DATABASE(,"Expression","User Specified","FSC na stacji 3");

REPLICATE, 10,,SecondsToBaseTime(54000),Yes,Yes,,24,Seconds,No,No,,Yes;

ENTITIES: Entity 1,Picture.Box,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
Appendix 6. Fast moving products zone, U-shape, 4 station in raw arrangement documentation (Siman language)

Model statements for module: Create 1
CREATE module creates the entity. The entity is created with between arrival time 5.5 seconds, with exponential probability distribution.

```siman
74$ CREATE, 1,SecondstoBaseTime(0.0),Order:SecondstoBaseTime(EXPO(5.5)):NEXT(75$);
75$ ASSIGN: Start point.NumberOut=Start point.NumberOut + 1:NEXT(11$);
```

Model statements for module: Assign 20
ASSIGN module assigns to variable 2 new value. When entity is passing through assign module, to variable 2 is added value of 1. The variable simulates the length of the line before station.

```siman
11$ ASSIGN: Variable 2=variable 2+1:NEXT(7$);
```

Model statements for module: Decide 3
DECIDE module checks whether the picker is busy or idle. If the picker 1 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.

```siman
7$ BRANCH, 1:
    If,Variable 1==1,78$,Yes:
    Else,79$,Yes;
78$ ASSIGN: If picker1 busy.NumberOut True=If picker1 busy.NumberOut True + 1:NEXT(4$);
79$ ASSIGN: If picker1 busy.NumberOut False=If picker1 busy.NumberOut False + 1:NEXT(50$);
```

Model statements for module: Hold 1
HOLD modules stores entities before station 1 until signal with decelerated value appears from SIGNAL module. The HOLD module release one entity per one signal.

```siman
4$ QUEUE, Hold 1.Queue;
    WAIT: 1,1:NEXT(50$);
```

Model statements for module: Assign 29
ASSIGN module assigns four variables to the entity. Variable Counter 1 is responsible for counting how many items have been picked already. Variable Repeat 1 is responsible for determining number of items to be picked within the station. Variable 2 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 1 determines the state of picker (busy / idle).

```siman
50$ ASSIGN: Counter 1=0:
    Repeat 1=anint(pois(1.2)):
    Variable 2=variable 2-1:
    Variable 1=1:NEXT(53$);
```

Model statements for module: Record 1
RECORD module records data about number of product being picked in station 1 within each order.
PROCESS module simulates the activity of picking the picking list by the picker on station 1.

ASSIGN:  Picking list.NumberIn=Picking list.NumberIn + 1:  Picking list.WIP=Picking list.WIP+1;

QUEUE,  Picking list.Queue;

SEIZE,  3,VA:  Resource 1,1:NEXT(81$);

DELAY:  Triangular(0.5,1,1.5),,VA;

RELEASE:  Resource 1,1;

ASSIGN:  Picking list.NumberOut=Picking list.NumberOut + 1:  Picking list.WIP=Picking list.WIP-1:NEXT(0$);

DECIDE module checks weather the picker is busy or idle. If the picker 2 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.

BRANCH,  1:
With,75.5/100,8$,Yes:
With,19.64/100,9$,Yes:
With,0.87/100,10$,Yes:
Else,6$,Yes;

SIGNAL module sends the signal to hold module.

SIGNAL:  1:NEXT(73$);

RECORD module records data about time when the orders were passing station 1.

ASSIGN module assigns to variable 4 and 1 new value. Variable 1 determines state of the picker (busy / idle). When entity is passing through assign module, to variable 4 is added value of 1. The variable simulates the length of the line before station.

ASSIGN:  Variable 1=0:
Variable 4=variable 4+1:NEXT(17$);

DECIDE module checks weather the picker is busy or idle. If the picker 3 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.

BRANCH,  1:
If,Variable 3==1,133$,Yes:
Else,134$,Yes;

ASSIGN:  If picker2 busy.NumberOut True=If picker2 busy.NumberOut True + 1:NEXT(14$);
HOLD modules store entities before station 2 until signal with decelerated value appears from SIGNAL module. The HOLD module release one entity per one signal.

ASSIGN module assigns four variables to the entity. Variable Counter 2 is responsible for counting how many items have been picked already. Variable Repeat 2 is responsible for determining number of items to be picked within the station. Variable 4 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 3 determines the state of picker (busy / idle).

RECORD module records data about number of product being picked in station 2 within each order.

PROCESS module simulates the activity of picking the picking list by the picker on station 2.

DECIDE module determines from which group of the bins the order has product to be picked.

SIGNAL module sends the signal to hold module.
RECORD module records data about times when the order was passing through station 2.

ASSIGN module assigns to variable 6 and 3 new value. Variable 3 determines state of the picker (busy / idle). When entity is passing through assign module, to variable 6 is added value of 1. The variable simulates the length of the line before station.

DECIDE module checks whether the picker is busy or idle. If the picker 3 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.

ASSIGN module assigns four variables to the entity. Variable Counter 3 is responsible for counting how many items have been picked already. Variable Repeat 3 is responsible for determining number of items to be picked within the station. Variable 6 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 5 determines the state of picker (busy / idle).

RECORD module records data about number of product being picked in station 3 within each order.
193$ QUEUE, Picking list3.Queue;
192$ SEIZE, 3,VA:
    Resource 3,1:NEXT(191$);
191$ DELAY: Triangular(0.5,1,1.5),,VA;
190$ RELEASE: Resource 3,1;
238$ ASSIGN: Picking list3.NumberOut=Picking list3.NumberOut + 1:
    Picking list3.WIP=Picking list3.WIP-1:NEXT(23$);

; Model statements for module: Decide 6
DECIDE module determines from which group of the bins the order has product to be picked.
23$ BRANCH, 1:
    With,73.53/100,34$,Yes:
    With,21.53/100,30$,Yes:
    With,0.95/100,31$,Yes:
    Else,27$,Yes;

; Model statements for module: Signal 3
SIGNAL module sends the signal to hold module.
27$ SIGNAL: 3:NEXT(71$);

; Model statements for module: Record 10
RECORD module records data about times when orders were passing through station 3.
71$ TALLY: A3 total,TNOW,1:NEXT(29$);
; Model statements for module: Assign 25
ASSIGN module assigns to variable 8 new value. When entity is passing through
assign module, to variable 8 is added value of 1. The variable simulates the length of
the line before station.
29$ ASSIGN: Variable 8=variable 8+1:
    Variable 5=0:NEXT(43$);
; Model statements for module: Decide 9
DECIDE module checks weather the picker is busy or idle. If the picker 4 is idle order
is transported to the picker, other case it is stopped in the queue in HOLD module.
43$ BRANCH, 1:
    If,Variable 7==1,243$,Yes:
    Else,244$,Yes;
243$ ASSIGN: If picker4 busy.NumberOut True=If picker4
    busy.NumberOut True + 1:NEXT(40$);
244$ ASSIGN: If picker4 busy.NumberOut False=If picker4
    busy.NumberOut False + 1:NEXT(62$);
; Model statements for module: Hold 4
HOLD modules stores entities before station 4 until signal with decelerated value
appears from SIGNAL module. The HOLD module release one entity per one signal.
40$ QUEUE, Hold 4.Queue;
    WAIT: 4,1:NEXT(62$);
; Model statements for module: Assign 35
ASSIGN module assigns four variables to the entity. Variable Counter 4 is responsible for counting how many items have been picked already. Variable Repeat 4 is responsible for determining number of items to be picked within the station. Variable 8 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 7 determines the state of picker (busy / idle).

62$ ASSIGN: Variable 8=variable 8-1:
   Variable 7=1:
   Counter 4=0:
   Repeat 4=anint(pois(1.2)):NEXT(65$);
;
   Model statements for module: Record 4
RECORD module records data about number of product being picked in station 3 within each order.
65$ TALLY: FSC na stacji 4,Repeat 4,1:NEXT(39$);
;
   Model statements for module: Process 64
PROCESS module simulates the activity of picking the picking list by picker.
39$ ASSIGN: Picking list4.NumberIn=Picking list4.NumberIn + 1:
   Picking list4.WIP=Picking list4.WIP+1;

248$ QUEUE, Picking list4.Queue;
247$ SEIZE, Resource 4,1:NEXT(246$);

246$ DELAY: Triangular(0.5,1,1.5),,VA;
245$ RELEASE: Resource 4,1;
293$ ASSIGN: Picking list4.NumberOut=Picking list4.NumberOut + 1:
   Picking list4.WIP=Picking list4.WIP-1:NEXT(38$);
;
   Model statements for module: Decide 8
DECIDE module determines from which group of the bins the order has product to be picked.
38$ BRANCH, 1:
   With,73.4/100,48$,Yes:
   With,22.6/100,45$,Yes:
   With,0/100,46$,Yes:
   Else,42$,Yes;
;
   Model statements for module: Signal 4
SIGNAL module sends the signal to hold module.
;
42$ SIGNAL: 4:NEXT(44$);
;
   Model statements for module: Assign 28
44$ ASSIGN: Variable 7=0:NEXT(70$);
;
   Model statements for module: Record 9
RECORD module records data about times when orders were passing through station 4.
70$ TALLY: A4 total,TNOW,1:NEXT(2$);
;
   Model statements for module: Dispose 1
2$ ASSIGN: End point.NumberOut=End point.NumberOut + 1;
298$ DISPOSE: Yes;
PROCESS module simulates the activity of picking the product from hands area on station 4.

48$ ASSIGN:  Hands area4.NumberIn=Hands area4.NumberIn + 1:
           Hands area4.WIP=Hands area4.WIP+1;
302$ QUEUE,  Hands area4.Queue;
301$ SEIZE,   Hands area4.WIP=Hands area4.WIP+1:
           Resource 4,1:NEXT(300$);
300$ DELAY:   Triangular(0,0.2,0.4),,VA;
299$ RELEASE: Resource 4,1;
347$ ASSIGN:  Hands area4.NumberOut=Hands area4.NumberOut + 1:
           Hands area4.WIP=Hands area4.WIP-1:NEXT(47$);

PROCESS module simulates the activity of putting the products to the box.

47$ ASSIGN:  Put products into box4.NumberIn=Put products into box4.NumberIn + 1:
           Put products into box4.WIP=Put products into box4.WIP+1;
353$ QUEUE,  Put products into box4.Queue;
352$ SEIZE,   Resource 4,1:NEXT(351$);
351$ DELAY:   Triangular(1,1.4,2),,VA;
350$ RELEASE: Resource 4,1;
398$ ASSIGN:  Put products into box4.NumberOut=Put products into box4.NumberOut + 1:
           Put products into box4.WIP=Put products into box4.WIP-1:NEXT(69$);

PROCESS module simulates the activity of picking the product from one step area on station 4.

64$ BRANCH,   1:
           If,Counter 4<Repeat 4,401$,Yes:
           Else,402$,Yes;
401$ ASSIGN:  Repeat4.NumberOut True=Repeat4.NumberOut True + 1:NEXT(39$);
402$ ASSIGN:  Repeat4.NumberOut False=Repeat4.NumberOut False + 1:NEXT(42$);

PROCESS module simulates the activity of putting the products to the box.
ASSIGN: One step area4.NumberIn = One step area4.NumberIn + 1;
One step area4.WIP = One step area4.WIP + 1;
QUEUE, One step area4.Queue;
SEIZE, 2, VA:
Resource 4,1: NEXT(404$);
DELAY: Triangular(1.4, 2.2, 8),, VA;
RELEASE: Resource 4,1;
ASSIGN: One step area4.NumberOut = One step area4.NumberOut + 1;
One step area4.WIP = One step area4.WIP - 1:NEXT(49$);

; Model statements for module: Process 69
PROCESS module simulates the activity of turning around by picker 4.
ASSIGN: Turning back4.NumberIn = Turning back4.NumberIn + 1:
Turning back4.WIP = Turning back4.WIP + 1;
QUEUE, Turning back4.Queue;
SEIZE, 2, VA:
Resource 4,1: NEXT(455$);
DELAY: Triangular(1, 2, 3),, VA;
RELEASE: Resource 4,1;
ASSIGN: Turning back4.NumberOut = Turning back4.NumberOut + 1:
Turning back4.WIP = Turning back4.WIP - 1:NEXT(47$);

; Model statements for module: Process 66
PROCESS module simulates the activity of picking products from two steps area on station 4.
ASSIGN: Two steps area4.NumberIn = Two steps area4.NumberIn + 1:
Two steps area4.WIP = Two steps area4.WIP + 1;
QUEUE, Two steps area4.Queue;
SEIZE, 2, VA:
Resource 4,1: NEXT(506$);
DELAY: Triangular(3, 4, 4.8),, VA;
RELEASE: Resource 4,1;
ASSIGN: Two steps area4.NumberOut = Two steps area4.NumberOut + 1:
Two steps area4.WIP = Two steps area4.WIP - 1:NEXT(49$);

; Model statements for module: Process 60
PROCESS module simulates the activity of picking product in hands area on station 3.
ASSIGN: Hands area3.NumberIn = Hands area3.NumberIn + 1:
Hands area3.WIP = Hands area3.WIP + 1;
QUEUE, Hands area3.Queue;
SEIZE, 2, VA:
Resource 3,1: NEXT(557$);
DELAY: Triangular(0, 0.2, 0.4),, VA;
RELEASE: Resource 3,1;
ASSIGN: Hands area3.NumberOut = Hands area3.NumberOut + 1:
Hands area3.WIP = Hands area3.WIP - 1:NEXT(33$);

; Model statements for module: Process 59
PROCESS module simulates the activity of putting products into the box.
ASSIGN: Put products into box3.NumberIn=Put products into box3.NumberIn + 1:
    Put products into box3.WIP=Put products into box3.WIP+1;

QUEUE, Put products into box3.Queue;

SEIZE, 2, VA:
    Resource 3,1:NEXT(608$);

DELAY: Triangular(1,1.4,2), VA;

RELEASE: Resource 3,1;

ASSIGN: Put products into box3.NumberOut=Put products into box3.NumberOut + 1:
    Put products into box3.WIP=Put products into box3.WIP-1:NEXT(68$);

RECORD module counts number of boxes that are passing through the station 3.
COUNT: Passed3,1:NEXT(58$);

ASSIGN module assigns to counter 3 new value. When entity is passing through assign module, to counter 3 is added value of 1. The variable simulates the number of products already picked within the station.

ASSIGN: Counter 3=Counter 3+1:NEXT(59$);

BRANCH, 1:
    If, Counter 3<Repeat 3,658$, Yes:
    Else, 659$, Yes;

ASSIGN: Repeat3.NumberOut True=Repeat3.NumberOut True + 1:NEXT(24$);

ASSIGN: Repeat3.NumberOut False=Repeat3.NumberOut False + 1:NEXT(27$);

PROCESS module simulates the activity of picking product in one step area on station 3.
ASSIGN: One step area3.NumberIn=One step area3.NumberIn + 1:
    One step area3.WIP=One step area3.WIP+1;

QUEUE, One step area3.Queue;

SEIZE, 2, VA:
    Resource 3,1:NEXT(661$);

DELAY: Triangular(1.4,2.2,8), VA;

RELEASE: Resource 3,1;

ASSIGN: One step area3.NumberOut=One step area3.NumberOut + 1:
    One step area3.WIP=One step area3.WIP-1:NEXT(37$);

PROCESS module simulates the activity of turning back by picker number 3.
ASSIGN: Turning back3.NumberIn=Turning back3.NumberIn + 1:
    Turning back3.WIP=Turning back3.WIP+1;

QUEUE, Turning back3.Queue;
PROCESS module simulates the activity of picking product in two step area on station 3.

ASSIGN: Two steps area3.NumberIn=Two steps area3.NumberIn + 1:
Two steps area3.WIP=Two steps area3.WIP+1;

QUEUE, Two steps area3.Queue;
SEIZE, 2,VA:
Resource 3,1:NEXT(763$);
DELAY: Triangular(3,4,4.8),,VA;
RELEASE: Resource 3,1;
ASSIGN: Two steps area3.NumberOut=Two steps area3.NumberOut + 1:
Two steps area3.WIP=Two steps area3.WIP-1:NEXT(37$);

; Model statements for module: Process 43
PROCESS module simulates the activity of picking product in hands area on station 2.

ASSIGN: Hands area2.NumberIn=Hands area2.NumberIn + 1:
Hands area2.WIP=Hands area2.WIP+1;

QUEUE, Hands area2.Queue;
SEIZE, 2,VA:
Resource 2,1:NEXT(814$);
DELAY: Triangular(0,0.2,0.4),,VA;
RELEASE: Resource 2,1;
ASSIGN: Hands area2.NumberOut=Hands area2.NumberOut + 1:
Hands area2.WIP=Hands area2.WIP-1:NEXT(21$);

; Model statements for module: Process 42
PROCESS module simulates the activity of dropping the products into the box on station 2.

ASSIGN: Put products into box2.NumberIn=Put products into box2.NumberIn + 1:
Put products into box2.WIP=Put products into box2.WIP+1;

QUEUE, Put products into box2.Queue;
SEIZE, 2,VA:
Resource 2,1:NEXT(865$);
DELAY: Triangular(1,1.4,2),,VA;
RELEASE: Resource 2,1;
ASSIGN: Put products into box2.NumberOut=Put products into box2.NumberOut + 1:
Put products into box2.WIP=Put products into box2.WIP-1:NEXT(67$);

; Model statements for module: Record 6
RECORD module counts the number of boxes that passing through the station 2.

67$ COUNT: Passed2,1:NEXT(56$);

; Model statements for module: Assign 33

ASSIGN module assigns to counter 3 new value. When entity is passing through
assign module, to counter 3 is added value of 1. The variable simulates the number of
products already picked within the station.

56$ ASSIGN: Counter 2=Counter 2+1:NEXT(57$);

; Model statements for module: Decide 11

57$ BRANCH, 1:
    If,Counter 2<Repeat 2,915$,Yes:
    Else,916$,Yes;

915$ ASSIGN: Repeat2.NumberOut True=Repeat2.NumberOut True +
1:NEXT(13$);

916$ ASSIGN: Repeat2.NumberOut False=Repeat2.NumberOut False +
1:NEXT(16$);

; Model statements for module: Process 35

PROCESS module simulates the activity of picking product in one step area on station
2.

18$ ASSIGN: One step area2.NumberIn=One step area2.NumberIn + 1:
    One step area2.WIP=One step area2.WIP+1;

920$ QUEUE, One step area2.Queue;

919$ SEIZE, 2,VA:
    Resource 2,1:NEXT(918$);

918$ DELAY: Triangular(1.4,2,2.8),,VA;

917$ RELEASE: Resource 2,1;

965$ ASSIGN: One step area2.NumberOut=One step area2.NumberOut + 1:
    One step area2.WIP=One step area2.WIP-1:NEXT(36$);

; Model statements for module: Process 62

PROCESS module simulates the activity of turning around by picker number 2.

36$ ASSIGN: Turning back2.NumberIn=Turning back2.NumberIn + 1:
    Turning back2.WIP=Turning back2.WIP+1;

971$ QUEUE, Turning back2.Queue;

970$ SEIZE, 2,VA:
    Resource 2,1:NEXT(969$);

969$ DELAY: Triangular(1,2,3),,VA;

968$ RELEASE: Resource 2,1;

1016$ ASSIGN: Turning back2.NumberOut=Turning back2.NumberOut + 1:
    Turning back2.WIP=Turning back2.WIP-1:NEXT(21$);

; Model statements for module: Process 36

PROCESS module simulates the activity of picking product in two step area on station
2.

19$ ASSIGN: Two steps area2.NumberIn=Two steps area2.NumberIn + 1:
    Two steps area2.WIP=Two steps area2.WIP+1;

1022$ QUEUE, Two steps area2.Queue;

1021$ SEIZE, 2,VA:
Resource 2,1:NEXT(1020$);
1020$ DELAY: Triangular(3,4,4.8),,VA;
1019$ RELEASE: Resource 2,1;
1067$ ASSIGN: Two steps area2.NumberOut=Two steps area2.NumberOut + 1:
Two steps area2.WIP=Two steps area2.WIP-1:NEXT(36$);
; Model statements for module: Process 19
PROCESS module simulates the activity of picking product in hands area on station 1.
8$ ASSIGN: Hands area1.NumberIn=Hands area1.NumberIn + 1:
Hands area1.WIP=Hands area1.WIP+1;
1073$ QUEUE, Hands area1.Queue;
1072$ SEIZE, 2,VA:
Resource 1,1:NEXT(1071$);
1071$ DELAY: Triangular(0,0.2,0.4),,VA;
1070$ RELEASE: Resource 1,1;
1118$ ASSIGN: Hands area1.NumberOut=Hands area1.NumberOut + 1:
Hands area1.WIP=Hands area1.WIP-1:NEXT(1$);
; Model statements for module: Process 2
PROCESS module simulates the activity of dropping the products to the box by the picker 1.
1$ ASSIGN: Put products into box1.NumberIn=Put products into box1.NumberIn + 1:
Put products into box1.WIP=Put products into box1.WIP+1;
1124$ QUEUE, Put products into box1.Queue;
1123$ SEIZE, 2,VA:
Resource 1,1:NEXT(1122$);
1122$ DELAY: Triangular(1,1.4,2),,VA;
1121$ RELEASE: Resource 1,1;
1169$ ASSIGN: Put products into box1.NumberOut=Put products into box1.NumberOut + 1:
Put products into box1.WIP=Put products into box1.WIP-1:NEXT(66$);
; Model statements for module: Record 5
RECORD module counts the number of boxes that passing through the station 1.
66$ COUNT: Passed1,1:NEXT(51$);
; Model statements for module: Assign 30
ASSIGN module assigns to counter 1 new value. When entity is passing through assign module, to counter 1 is added value of 1. The variable simulates the number of products already picked within the station.
51$ ASSIGN: Counter 1=Counter 1+1:NEXT(52$);
; Model statements for module: Decide 10
52$ BRANCH, 1:
If,Counter 1<Repeat 1,1172$,Yes:
Else,1173$,Yes;
ASSIGN: Repeat1.NumberOut True = Repeat1.NumberOut True + 1:NEXT(3$);
ASSIGN: Repeat1.NumberOut False = Repeat1.NumberOut False + 1:NEXT(6$);

Model statements for module: Process 20
PROCESS module simulates the activity of picking product in one step area on station 1.

ASSIGN: One step area1.NumberIn = One step area1.NumberIn + 1:
        One step area1.WIP = One step area1.WIP + 1;
QUEUE, One step area1.Queue;
SEIZE, 2,VA:
        Resource 1,1:NEXT(1175$);
DELAY: Triangular(1.4,2,2.8),,VA;
RELEASE: Resource 1,1;
ASSIGN: One step area1.NumberOut = One step area1.NumberOut + 1:
        One step area1.WIP = One step area1.WIP - 1:NEXT(35$);

Model statements for module: Process 61
PROCESS module simulates the activity of turning back by the picker number 1.

ASSIGN: Turning back1.NumberIn = Turning back1.NumberIn + 1:
        Turning back1.WIP = Turning back1.WIP + 1;
QUEUE, Turning back1.Queue;
SEIZE, 2,VA:
        Resource 1,1:NEXT(1226$);
DELAY: Triangular(1,2,3),,VA;
RELEASE: Resource 1,1;
ASSIGN: Turning back1.NumberOut = Turning back1.NumberOut + 1:
        Turning back1.WIP = Turning back1.WIP - 1:NEXT(1$);

Model statements for module: Process 21
PROCESS module simulates the activity of picking product in two step area on station 1.

ASSIGN: Two steps area1.NumberIn = Two steps area1.NumberIn + 1:
        Two steps area1.WIP = Two steps area1.WIP + 1;
QUEUE, Two steps area1.Queue;
SEIZE, 2,VA:
        Resource 1,1:NEXT(1277$);
DELAY: Triangular(3,4,4.8),,VA;
RELEASE: Resource 1,1;
ASSIGN: Two steps area1.NumberOut = Two steps area1.NumberOut + 1:
        Two steps area1.WIP = Two steps area1.WIP - 1:NEXT(35$);

PROJECT, "Unnamed Project","xxx",,,No,Yes,Yes,Yes,No,No,No,No,No;
VARIABLES: Picking

list3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Two steps area1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Hands area1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Turning back1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Turning back3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  End point.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Put products into

box1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Put products into

box3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  One step area1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  If picker4 busy.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):  
  Picking list.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  One step area4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Picking list.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  One step area1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Put products into

box3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Turning back2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Put products into box4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Repeat1.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):  
  Picking list2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Repeat2.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):  
  One step area2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Hands area1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Two steps area3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  One step area4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Hands area3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  Hands area3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Picking list3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
  If picker3 busy.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):  
  Hands area4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Turning back3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  Two steps area3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
  If picker3 busy.NumberOut

True,CLEAR(Statistics),CATEGORY("Exclude"):  
  If picker2 busy.NumberOut

False,CLEAR(Statistics),CATEGORY("Exclude"):  
  One step area3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Turning back4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Picking list4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
  Repeat3.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):  

179
Hands area1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Put products into box1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  Two steps area1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Two steps area3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  One step area3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Repeat2.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
  Picking list2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  If picker1 busy.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
  Counter 1,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Counter 2,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Counter 3,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Counter 4,CLEAR(System),CATEGORY("User Specified-User Specified"):
  If picker4 busy.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
  Put products into
box2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Turning back1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  Repeat 1,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Repeat 2,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Put products into box3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  Repeat 3,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Repeat 4,CLEAR(System),CATEGORY("User Specified-User Specified"):
  Turning back2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Turning back2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Turning back4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Put products into
box2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Put products into
box4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Repeat4.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
  Two steps area2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  Hands area2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  Hands area3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Two steps area2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
  Repeat3.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
  If picker1 busy.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
  One step area2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  One step area1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Turning back3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
  One step area3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Hands area2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
  Hands area4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
Picking list4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Picking list3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Picking list2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude");
Picking list4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude");
One step area2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Put products into
box1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Two steps area4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Hands area4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Put products into
box4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Turning back4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Start point.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Two steps area2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude");
Repeat1.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude");
Two steps area4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude");
One step area4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Variable 1,CLEAR(System),CATEGORY("User Specified-User Specified");
Variable 2,CLEAR(System),CATEGORY("User Specified-User Specified");
Variable 3,CLEAR(System),CATEGORY("User Specified-User Specified");
Variable 4,CLEAR(System),CATEGORY("User Specified-User Specified");
Picking list.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Variable 5,CLEAR(System),CATEGORY("User Specified-User Specified");
If picker2 busy.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude");
Variable 6,CLEAR(System),CATEGORY("User Specified-User Specified");
Variable 7,CLEAR(System),CATEGORY("User Specified-User Specified");
Variable 8,CLEAR(System),CATEGORY("User Specified-User Specified");
Put products into box2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude");
Hands area2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Turning back1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Two steps area1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
Repeat4.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude");
Two steps area4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");

QUEUES: One step area4.Queue,FIFO,,AUTOSTATS(Yes,..):
Turning back1.Queue,FIFO,,AUTOSTATS(Yes,..):
Picking list.Queue,FIFO,,AUTOSTATS(Yes,..):
Hold 3.Queue,FIFO,,AUTOSTATS(Yes,..):
Biny 2.Queue,FIFO,,AUTOSTATS(Yes,..):
Hands area4.Queue,FIFO,,AUTOSTATS(Yes,..):
Put products into box2.Queue,FIFO,,AUTOSTATS(Yes,..):
Hold 4.Queue,FIFO,,AUTOSTATS(Yes,..):
Biny 3.Queue,FIFO,,AUTOSTATS(Yes,..):
Turning back2.Queue,FIFO,,AUTOSTATS(Yes,..):
Put products into box3.Queue,FIFO,,AUTOSTATS(Yes,):
Two steps area1.Queue,FIFO,,AUTOSTATS(Yes,):
Biny 4.Queue,FIFO,,AUTOSTATS(Yes,):
One step area1.Queue,FIFO,,AUTOSTATS(Yes,):
Turning back3.Queue,FIFO,,AUTOSTATS(Yes,):
Picking list2.Queue,FIFO,,AUTOSTATS(Yes,):
Hands area1.Queue,FIFO,,AUTOSTATS(Yes,):
Two steps area2.Queue,FIFO,,AUTOSTATS(Yes,):
Put products into box4.Queue,FIFO,,AUTOSTATS(Yes,):
One step area2.Queue,FIFO,,AUTOSTATS(Yes,):
Turning back4.Queue,FIFO,,AUTOSTATS(Yes,):
Hold 1.Queue,FIFO,,AUTOSTATS(Yes,):
biny 0.Queue,FIFO,,AUTOSTATS(Yes,):
Picking list3.Queue,FIFO,,AUTOSTATS(Yes,):
Biny 5.Queue,FIFO,,AUTOSTATS(Yes,):
Hands area2.Queue,FIFO,,AUTOSTATS(Yes,):
Two steps area3.Queue,FIFO,,AUTOSTATS(Yes,):
Biny 6.Queue,FIFO,,AUTOSTATS(Yes,):
One step area3.Queue,FIFO,,AUTOSTATS(Yes,):
Two steps area4.Queue,FIFO,,AUTOSTATS(Yes,):
Hold 2.Queue,FIFO,,AUTOSTATS(Yes,):
Biny 1.Queue,FIFO,,AUTOSTATS(Yes,):
Picking list4.Queue,FIFO,,AUTOSTATS(Yes,):
Hands area3.Queue,FIFO,,AUTOSTATS(Yes,):
Put products into box1.Queue,FIFO,,AUTOSTATS(Yes,):
Biny 7.Queue,FIFO,,AUTOSTATS(Yes,);

PICTURES:
Picture.Airplane:
Picture.Green Ball:
Picture.Blue Page:
Picture.Telephone:
Picture.Blue Ball:
Picture.Yellow Page:
Picture.EMail:
Picture.Yellow Ball:
Picture.Bike:
Picture.Report:
Picture.Van:
Picture.Widgets:
Picture.Envelope:
Picture.Fax:
Picture.Truck:
Picture.Letter:
Picture.Box:
Picture.Woman:
Picture.Package:
FAILURES:  Failure 1, Time(EXPO(3000), EXPO(20),);  
            Failure 2, Time(EXPO(3000), EXPO(20),);  
            Failure 3, Time(EXPO(3000), EXPO(20),);  
            Failure 4, Time(EXPO(3000), EXPO(20),);  

RESOURCES:  Resource  
            1, Capacity(1),,, COST(0.0,0.0,0.0,0.0), CATEGORY(Resources), FAILURE(Failure 1, Preempt), AUTOSTATS(Yes,);  
            Resource 2, Capacity(1),,, COST(0.0,0.0,0.0,0.0), CATEGORY(Resources), FAILURE(Failure 2, Preempt), AUTOSTATS(Yes,);  
            Resource 3, Capacity(1),,, COST(0.0,0.0,0.0,0.0), CATEGORY(Resources), FAILURE(Failure 3, Preempt), AUTOSTATS(Yes,);  
            Resource 4, Capacity(1),,, COST(0.0,0.0,0.0,0.0), CATEGORY(Resources), FAILURE(Failure 4, Preempt), AUTOSTATS(Yes,);  

COUNTERS:  Passed1,,DATABASE(,"Count","User Specified","Passed1");  
            Passed2,,DATABASE(,"Count","User Specified","Passed2");  
            Passed3,,DATABASE(,"Count","User Specified","Passed3");  
            Passed4,,DATABASE(,"Count","User Specified","Passed4");  

TALLYES:  A4 total,  
            "C:\Documents and Settings\Administrator\Pulpit\magisterka\A\Segment A 4 stacje\sorting solution\A4 tnow.dat",  
            DATABASE(,"Expression","User Specified","A4 total");  
            A1 total,  
            "C:\Documents and Settings\Administrator\P pulpit\magisterka\A\Segment A 4 stacje\sorting solution\A1 tnow.dat",  
            DATABASE(,"Expression","User Specified","A1 total");  
            FSC na stacje 1,,DATABASE(,"Expression","User Specified","FSC na stacje 1");  
            A2 total,  
            "C:\Documents and Settings\Administrator\Pulpit\magisterka\A\Segment A 4 stacje\sorting solution\A2 tnow.dat",  
            DATABASE(,"Expression","User Specified","A2 total");  
            FSC na stacje 2,,DATABASE(,"Expression","User Specified","FSC na stacje 2");
FSC na stację 3, DATABASE("Expression", "User Specified", "FSC na stację 3");
FSC na stację 4, DATABASE("Expression", "User Specified", "FSC na stację 4");
A3 total,
"C:\Documents and Settings\Administrator\Pulpit\magisterka\A\Segment A 4 stacje\sorting solution\A3 tnow.dat",
DATABASE("Expression", "User Specified", "A3 total");
REPLICATE, 10, SecondsToBaseTime(54000), Yes, Yes,, 24, Seconds, No, No,, Yes;
ENTITIES: Order, Picture, Report, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, AUTOSTATS(Yes,);
Appendix 7. The horizontal, single station layout with diagonal storage policy (Siman language)

Model statements for module: Create 1
CREATE module creates the entity. The entity is created with between arrival time 13,5 seconds, with exponential probability distribution.
;
49$ CREATE, 1,SecondstoBaseTime(0.0),Entity
1:SecondstoBaseTime(EXPO(13.4)):NEXT(50$);

50$ ASSIGN: Start.NumberOut=Start.NumberOut + 1:NEXT(7$);

; ; ; Model statements for module: Assign 20
ASSIGN module assigns to variable 2 new value. When entity is passing through assign module, to variable 2 is added value of 1. The variable simulates the length of the line before station.
;
7$ ASSIGN: Variable 2=variable 2+1:NEXT(5$);

; ; ; Model statements for module: Decide 3
DECIDE module checks weather the picker is busy or idle. If the picker 1 is idle order is transported to the picker, other case it is stopped in the queue in HOLD module.
;
5$ BRANCH, 1:
    If,Variable 1==1,53$,Yes:
    Else,54$,Yes;
53$ ASSIGN: If picker 1 busy.NumberOut True=If picker 1 busy.NumberOut True + 1:NEXT(2$);

54$ ASSIGN: If picker 1 busy.NumberOut False=If picker 1 busy.NumberOut False + 1:NEXT(6$);

; ; ; Model statements for module: Hold 1
HOLD modules stores entities before station 1 until signal with decelerated value appears from SIGNAL module. The HOLD module release one entity per one signal.
;
$\text{QUEUE, Hold 1.Queue;}
\text{WAIT: 1,1:NEXT(6$);}$

; ;
; ; Model statements for module: Assign 1
\text{ASSIGN module assigns four variables to the entity. Variable Licznik 1 is responsible for counting how many items have been picked already. Variable Repeat 1 is responsible for determining number of items to be picked within the station. Variable 2 is responsible for determining the present number of orders being in the queue in HOLD module. Variable 1 determines the state of picker (busy / idle). ;}
$\text{6$ ASSIGN: Licznik 1=0: Repeat 1=anint(pois(0.24)): Variable 2=variable 2-1: Variable 1=1:NEXT(11$);}$

; ;
; ; Model statements for module: Process 64
\text{PROCESS module simulates the activity of picking the picking list on station 1. ;}
$\text{11$ ASSIGN: Picking picking list.NumberIn=Picking picking list.NumberIn + 1: Picking picking list.WIP=Picking picking list.WIP+1; 58$ QUEUE, Picking picking list.Queue; 57$ SEIZE, 3,VA: Resource 1,1:NEXT(56$);}$

$\text{56$ DELAY: Triangular(0.5,1,1.5),VA; 55$ RELEASE: Resource 1,1; 103$ ASSIGN: Picking picking list.NumberOut=Picking picking list.NumberOut + 1: Picking picking list.WIP=Picking picking list.WIP-1:NEXT(26$);}$

; ;
; ; Model statements for module: Record 2
\text{RECORD module counts the number of boxes that passing through the slow moving products station 1. ;}
$\text{26$ COUNT: C1,1:NEXT(13$);}$
Model statements for module: Decide 15
DECIDE module determines from which group of the bins the order has product to be picked.

13$   BRANCH,   1:
    With,64.14/100,16$,Yes:
    With,32.51/100,17$,Yes:
    Else,18$,Yes;

Model statements for module: Decide 18
DECIDE module determines from which group of the bins the order has product to be picked.

18$   BRANCH,   1:
    With,20.26/100,38$,Yes:
    With,15.91/100,39$,Yes:
    With,25.02/100,40$,Yes:
    With,18.16/100,41$,Yes:
    With,12.72/100,42$,Yes:
    With,7.92/100,46$,Yes:
    Else,43$,Yes;

Model statements for module: Process 140
PROCESS module simulates the walk of picker to the third raw in slow moving zone.

43$   ASSIGN:   Third raw6.NumberIn=Third raw6.NumberIn + 1:
            Third raw6.WIP=Third raw6.WIP+1;
113$   QUEUE,   Third raw6.Queue;
112$   SEIZE,   2,VA:
        Resource 1,1:NEXT(111$);
111$   DELAY:   Triangular(7,8,8.8),VA;
110$   RELEASE: Resource 1,1;
158$   ASSIGN:   Third raw6.NumberOut=Third raw6.NumberOut + 1:
            Third raw6.WIP=Third raw6.WIP-1:NEXT(23$);
PROCESS module simulates the activity of picking product in six step area.

23$ ASSIGN: Six steps area.NumberIn=Six steps area.NumberIn + 1:
Six steps area.WIP=Six steps area.WIP+1;
164$ QUEUE, Six steps area.Queue;
163$ SEIZE, 2,VA:
Resource 1,1:NEXT(162$);
162$ DELAY: Triangular(11,12,12.8),,VA;
161$ RELEASE: Resource 1,1;
209$ ASSIGN: Six steps area.NumberOut=Six steps area.NumberOut + 1:
Six steps area.WIP=Six steps area.WIP-1:NEXT(12$);

PROCESS module simulates the activity of turning around by the picker.

12$ ASSIGN: Turning around.NumberIn=Turning around.NumberIn + 1:
Turning around.WIP=Turning around.WIP+1;
215$ QUEUE, Turning around.Queue;
214$ SEIZE, 2,VA:
Resource 1,1:NEXT(213$);
213$ DELAY: Triangular(1,2,3),,VA;
212$ RELEASE: Resource 1,1;
260$ ASSIGN: Turning around.NumberOut=Turning around.NumberOut + 1:
Turning around.WIP=Turning around.WIP-1:NEXT(0$);

PROCESS module simulates the activity of dropping the products into the box.

0$ ASSIGN: Dropping products into the box.NumberIn=Dropping
products into the box.NumberIn + 1:
Dropping products into the box.WIP=Dropping products into the
box.WIP+1;
266$ QUEUE, Dropping products into the box.Queue;
265$ SEIZE, 2,VA:
Resource 1,1:NEXT(264$);
264$ DELAY: Triangular(1,1.4,2),,VA;
263$ RELEASE: Resource 1,1;
311$ ASSIGN: Dropping products into the box.NumberOut=Dropping products into the box.NumberOut + 1:
                Dropping products into the box.WIP=Dropping products into the box.WIP-1:NEXT(9$);

; ;
; ;  Model statements for module: Assign 27
ASSIGN module assigns to licznik 1 new value. When entity is passing through assign module, to licznik 1 is added value of 1. The variable simulates the number of products already picked within the station.
; 9$ ASSIGN: Licznik 1=Licznik 1+1:NEXT(10$);

; ;
; ;  Model statements for module: Decide 8
DECIDE module checks whether all products from particular station have been picked.
; 10$ BRANCH, 1:
                If,Licznik 1<Repeat 1,314$,$Yes:
                Else,315$,$Yes;
314$ ASSIGN: If all products has been packed.NumberOut True=If all products has been packed.NumberOut True + 1
                :NEXT(11$);
315$ ASSIGN: If all products has been packed.NumberOut False=If all products has been packed.NumberOut False + 1
                :NEXT(4$);

; ;
; ;  Model statements for module: Signal 1
SIGNAL module sends the signal to hold module.
; 4$ SIGNAL: 1:NEXT(48$);
; Model statements for module: Record 3
RECORD module record the times when order is going out of the station 1 in slow moving products zone.

; 48$         TALLY:     C1 total, TNOW, 1:NEXT(8$);

; ;
; ; Model statements for module: Assign 23
ASSIGN module assigns to the entity the state of the picker (idle)
; 8$ ASJIN: Variable 1 = 0: NEXT(1$);

; ;
; ; Model statements for module: Dispose 1
DISPOSE module – the end of the system.
; 1$ ASJIN: Dispose 1.NumberOut = Dispose 1.NumberOut + 1;
316$ DISPOSE: Yes;

; ;
; ; Model statements for module: Process 134
PROCESS module simulates the walk of the picker to the first raw of the racks.
; 38$ ASJIN: First raw7.NumberIn = First raw7.NumberIn + 1;
    First raw7.WIP = First raw7.WIP + 1;
320$ QUEUE, First raw7.Queue;
319$ SEIZE, 2, VA:
    Resource 1, 1: NEXT(318$);
318$ DELAY: Triangular(.1, .05, 1), VA;
317$ RELEASE: Resource 1, 1;
365$ ASJIN: First raw7.NumberOut = First raw7.NumberOut + 1;
    First raw7.WIP = First raw7.WIP - 1: NEXT(24$);

; ;
; ; Model statements for module: Process 119
PROCESS module simulates the activity of picking product in seven step area.
ASSIGN: Seven steps area.NumberIn=Seven steps area.NumberIn + 1:
Seven steps area.WIP=Seven steps area.WIP+1;

QUEUE, Seven steps area.Queue;
SEIZE, 2,VA:
Resource 1,1:NEXT(369$);

DELAY: Triangular(13,14,14.8),,VA;
RELEASE: Resource 1,1;
ASSIGN: Seven steps area.NumberOut=Seven steps area.NumberOut + 1:
Seven steps area.WIP=Seven steps area.WIP-1:NEXT(12$);

; ; ; Model statements for module: Process 136
PROCESS module simulates the walk of the picker to the first raw of the racks.
; ;
ASSIGN: First raw8.NumberIn=First raw8.NumberIn + 1:
First raw8.WIP=First raw8.WIP+1;
QUEUE, First raw8.Queue;
SEIZE, 2,VA:
Resource 1,1:NEXT(420$);

DELAY: Triangular(.1,0.5,1),,VA;
RELEASE: Resource 1,1;
ASSIGN: First raw8.NumberOut=First raw8.NumberOut + 1:
First raw8.WIP=First raw8.WIP-1:NEXT(47$);

; ; ; Model statements for module: Process 144
PROCESS module simulates the activity of picking product in eight step area.
; ;
ASSIGN: Eight steps area.NumberIn=Eight steps area.NumberIn + 1:
Eight steps area.WIP=Eight steps area.WIP+1;
QUEUE, Eight steps area.Queue;
SEIZE, 2,VA:
Resource 1,1:NEXT(471$);

DELAY: Triangular(13,14,14.8),,VA;
RELEASE: Resource 1,1;
ASSIGN: Eight steps area.NumberOut=Eight steps area.NumberOut + 1:
Eight steps area.WIP=Eight steps area.WIP-1:NEXT(12$);
; ; Model statements for module: Process 137
PROCESS module simulates the walk of the picker to the second raw of the racks.
;
40$ ASSIGN: Second raw6.NumberIn=Second raw6.NumberIn + 1:
Second raw6.WIP=Second raw6.WIP+1;
524$ QUEUE, Second raw6.Queue;
523$ SEIZE, 2,VA:
    Resource 1,1:NEXT(522$);
522$ DELAY: Triangular(3,4,4.8),,VA;
521$ RELEASE: Resource 1,1;
Second raw6.WIP=Second raw6.WIP-1:NEXT(23$);

; ; Model statements for module: Process 138
PROCESS module simulates the walk of the picker to the second raw of the racks.
;
41$ ASSIGN: Second raw7.NumberIn=Second raw7.NumberIn + 1:
Second raw7.WIP=Second raw7.WIP+1;
575$ QUEUE, Second raw7.Queue;
574$ SEIZE, 2,VA:
    Resource 1,1:NEXT(573$);
573$ DELAY: Triangular(3,4,4.8),,VA;
572$ RELEASE: Resource 1,1;
620$ ASSIGN: Second raw7.NumberOut=Second raw7.NumberOut + 1:
Second raw7.WIP=Second raw7.WIP-1:NEXT(24$);

; ; Model statements for module: Process 139
PROCESS module simulates the walk of the picker to the second raw of the racks.
;
42$ ASSIGN: Second raw8.NumberIn=Second raw8.NumberIn + 1:
Second raw8.WIP=Second raw8.WIP+1;
626$ QUEUE, Second raw8.Queue;
625$ SEIZE, 2,VA:
    Resource 1,1:NEXT(624$);
Model statements for module: Process 143

PROCESS module simulates the walk of the picker to the third row of the racks.

; 46$ ASSIGN: Third raw5.NumberIn=Third raw5.NumberIn + 1:
Third raw5.WIP=Third raw5.WIP+1;
677$ QUEUE, Third raw5.Queue;
676$ SEIZE, 2,VA:
Resource 1,1:NEXT(675$);
675$ DELAY: Triangular(7,8,8.8),,VA;
674$ RELEASE: Resource 1,1;
722$ ASSIGN: Third raw5.NumberOut=Third raw5.NumberOut + 1:
Third raw5.WIP=Third raw5.WIP-1:NEXT(22$);

Model statements for module: Process 117

PROCESS module simulates the activity of picking from five steps area.

; 22$ ASSIGN: Five steps area.NumberIn=Five steps area.NumberIn + 1:
Five steps area.WIP=Five steps area.WIP+1;
728$ QUEUE, Five steps area.Queue;
727$ SEIZE, 2,VA:
Resource 1,1:NEXT(726$);
726$ DELAY: Triangular(9,10,10.8),,VA;
725$ RELEASE: Resource 1,1;
773$ ASSIGN: Five steps area.NumberOut=Five steps area.NumberOut + 1:
Five steps area.WIP=Five steps area.WIP-1:NEXT(12$);

Model statements for module: Decide 16

DECIDE module determines from which group of the bins the order has product to be picked.
16$ BRANCH, 1:
   With, 16.22/100, 14$, Yes:
   With, 14.47/100, 27$, Yes:
   With, 13.11/100, 44$, Yes:
   With, 22.58/100, 28$, Yes:
   With, 18.28/100, 45$, Yes:
   Else, 15$, Yes;

;  Model statements for module: Process 113
PROCESS module simulates the walk to the third raw of the racks.
;
15$ ASSIGN: Third raw1.NumberIn=Third raw1.NumberIn + 1:
   Third raw1.WIP=Third raw1.WIP + 1;
781$ QUEUE, Third raw1.Queue;
780$ SEIZE, 2, VA:
      Resource 1, 1:NEXT(779$);
779$ DELAY: Triangular(7, 8, 8.8), VA;
778$ RELEASE: Resource 1, 1;
826$ ASSIGN: Third raw1.NumberOut=Third raw1.NumberOut + 1:
   Third raw1.WIP=Third raw1.WIP-1:NEXT(25$);

;  Model statements for module: Process 121
PROCESS module simulates the activity of picking in one step area.
;
25$ ASSIGN: One step area.NumberIn=One step area.NumberIn + 1:
   One step area.WIP=One step area.WIP + 1;
832$ QUEUE, One step area.Queue;
831$ SEIZE, 2, VA:
      Resource 1, 1:NEXT(830$);
830$ DELAY: Triangular(1, 2, 2.8), VA;
829$ RELEASE: Resource 1, 1;
877$ ASSIGN: One step area.NumberOut=One step area.NumberOut + 1:
   One step area.WIP=One step area.WIP-1:NEXT(12$);

;  Model statements for module: Process 111
PROCESS module simulates the walk to the first raw of the racks.

; 14$ ASSIGN: First raw1.NumberIn=First raw1.NumberIn + 1:
First raw1.WIP=First raw1.WIP+1;
883$ QUEUE, First raw1.Queue;
882$ SEIZE, 2,VA:
    Resource 1,1:NEXT(881$);
881$ DELAY: Triangular(.1,0.5,1),,VA;
880$ RELEASE: Resource 1,1;
928$ ASSIGN: First raw1.NumberOut=First raw1.NumberOut + 1:
First raw1.WIP=First raw1.WIP-1:NEXT(25$);

; ; Model statements for module: Process 122

PROCESS module simulates the walk to the first raw of the racks.

; 27$ ASSIGN: First raw2.NumberIn=First raw2.NumberIn + 1:
First raw2.WIP=First raw2.WIP+1;
934$ QUEUE, First raw2.Queue;
933$ SEIZE, 2,VA:
    Resource 1,1:NEXT(932$);
932$ DELAY: Triangular(.1,0.5,1),,VA;
931$ RELEASE: Resource 1,1;
979$ ASSIGN: First raw2.NumberOut=First raw2.NumberOut + 1:
First raw2.WIP=First raw2.WIP-1:NEXT(19$);

; ; Model statements for module: Process 114

PROCESS module simulates the activity of picking product from two steps area.

; 19$ ASSIGN: Two steps area.NumberIn=Two steps area.NumberIn + 1:
Two steps area.WIP=Two steps area.WIP+1;
985$ QUEUE, Two steps area.Queue;
984$ SEIZE, 2,VA:
    Resource 1,1:NEXT(983$);
983$ DELAY: Triangular(3.4,4.8),,VA;
982$ RELEASE: Resource 1,1;
1030$ ASSIGN: Two steps area.NumberOut=Two steps area.NumberOut + 1:
Two steps area.WIP=Two steps area.WIP-1:NEXT(12$);  

;  
;  
; Model statements for module: Process 141  
PROCESS module simulates the walk to the first raw of the racks.  
; 44$ ASSIGN: First raw3.NumberIn=First raw3.NumberIn + 1:  
First raw3.WIP=First raw3.WIP+1;  
1036$ QUEUE, First raw3.Queue;  
1035$ SEIZE, 2,VA:  
Resource 1,1:NEXT(1034$);  
1034$ DELAY: Triangular(.1,0.5,1),,VA;  
1033$ RELEASE: Resource 1,1;  
1081$ ASSIGN: First raw3.NumberOut=First raw3.NumberOut + 1:  
First raw3.WIP=First raw3.WIP-1:NEXT(20$);  

;  
;  
; Model statements for module: Process 115  
PROCESS module simulates the activity of picking products from three steps area.  
; 20$ ASSIGN: Three steps area.NumberIn=Three steps area.NumberIn + 1:  
Three steps area.WIP=Three steps area.WIP+1;  
1087$ QUEUE, Three steps area.Queue;  
1086$ SEIZE, 2,VA:  
Resource 1,1:NEXT(1085$);  
1085$ DELAY: Triangular(5,6,6.8),,VA;  
1084$ RELEASE: Resource 1,1;  
1132$ ASSIGN: Three steps area.NumberOut=Three steps area.NumberOut + 1:  
Three steps area.WIP=Three steps area.WIP-1:NEXT(12$);  

;  
;  
; Model statements for module: Process 123  
PROCESS module simulates the walk to the second raw of the racks.  
; 28$ ASSIGN: Second raw1.NumberIn=Second raw1.NumberIn + 1:  
Second raw1.WIP=Second raw1.WIP+1;  
1138$ QUEUE, Second raw1.Queue;
1137$ SEIZE, 2,VA:
   Resource 1,1:NEXT(1136$);

1136$ DELAY: Triangular(3,4,4.8),VA;
1135$ RELEASE: Resource 1,1;
1183$ ASSIGN: Second raw1.NumberOut=Second raw1.NumberOut + 1:
   Second raw1.WIP=Second raw1.WIP-1:NEXT(25$);

; ; Model statements for module: Process 142
PROCESS module simulates the walk to the second raw of the racks.
;
45$ ASSIGN: Second raw2.NumberIn=Second raw2.NumberIn + 1:
   Second raw2.WIP=Second raw2.WIP+1;
1189$ QUEUE, Second raw2.Queue;
1188$ SEIZE, 2,VA:
   Resource 1,1:NEXT(1187$);

1187$ DELAY: Triangular(3,4,4.8),VA;
1186$ RELEASE: Resource 1,1;
1234$ ASSIGN: Second raw2.NumberOut=Second raw2.NumberOut + 1:
   Second raw2.WIP=Second raw2.WIP-1:NEXT(19$);

; ; Model statements for module: Decide 17
DECIDE module determines from which group of the bins the order has product to be picked.
;
17$ BRANCH, 1:
   With,13.28/100,29$,Yes:
   With,12.32/100,30$,Yes:
   With,10.9/100,31$,Yes:
   With,18.02/100,32$,Yes:
   With,14.12/100,33$,Yes:
   With,11.13/100,34$,Yes:
   With,8.49/100,35$,Yes:
   With,6.65/100,36$,Yes:
   Else,37$,Yes;

; ; Model statements for module: Process 132
PROCESS module simulates the walk to the third raw of the racks.

```vdl
37$ ASSIGN:  Third raw4.NumberIn=Third raw4.NumberIn + 1:
            Third raw4.WIP=Third raw4.WIP+1;
1242$ QUEUE,  Third raw4.Queue;
1241$ SEIZE,  2, VA:
            Resource 1,1:NEXT(1240$);
1240$ DELAY:  Triangular(7,8,8.8), VA;
1239$ RELEASE: Resource 1,1;
1287$ ASSIGN:  Third raw4.NumberOut=Third raw4.NumberOut + 1:
            Third raw4.WIP=Third raw4.WIP-1:NEXT(21$);
```

.;
;
; Model statements for module: Process 116

PROCESS module simulates the activity of picking the products from four steps area.

```vdl
21$ ASSIGN:  Four steps area.NumberIn=Four steps area.NumberIn + 1:
            Four steps area.WIP=Four steps area.WIP+1;
1293$ QUEUE,  Four steps area.Queue;
1292$ SEIZE,  2, VA:
            Resource 1,1:NEXT(1291$);
1291$ DELAY:  Triangular(7,8,8.8), VA;
1290$ RELEASE: Resource 1,1;
1338$ ASSIGN:  Four steps area.NumberOut=Four steps area.NumberOut + 1:
            Four steps area.WIP=Four steps area.WIP-1:NEXT(12$);
```

.;
;
; Model statements for module: Process 124

PROCESS module simulates the walk to the first raw of the racks.

```vdl
29$ ASSIGN:  First raw4.NumberIn=First raw4.NumberIn + 1:
            First raw4.WIP=First raw4.WIP+1;
1344$ QUEUE,  First raw4.Queue;
1343$ SEIZE,  2, VA:
            Resource 1,1:NEXT(1342$);
1342$ DELAY:  Triangular(.1,0.5,1), VA;
1341$ RELEASE: Resource 1,1;
1389$ ASSIGN:  First raw4.NumberOut=First raw4.NumberOut + 1:
```

198
PROCESS module simulates the walk to the first raw of the racks.

ASSIGN: First raw5.NumberIn = First raw5.NumberIn + 1;
        First raw5.WIP = First raw5.WIP + 1;
QUEUE, First raw5.Queue;
SEIZE, Resource 1, 1:
DELAY: Triangular(.1, 0.5, 1), VA;
RELEASE: Resource 1, 1;
ASSIGN: First raw5.NumberOut = First raw5.NumberOut + 1;
        First raw5.WIP = First raw5.WIP - 1;

PROCESS module simulates the walk to the first raw of the racks.

ASSIGN: First raw6.NumberIn = First raw6.NumberIn + 1;
        First raw6.WIP = First raw6.WIP + 1;
QUEUE, First raw6.Queue;
SEIZE, Resource 1, 1:
DELAY: Triangular(.1, 0.5, 1), VA;
RELEASE: Resource 1, 1;
        First raw6.WIP = First raw6.WIP - 1;

PROCESS module simulates the walk to the second raw of the racks.

ASSIGN: Second raw3.NumberIn = Second raw3.NumberIn + 1;
        Second raw3.WIP = Second raw3.WIP + 1;
QUEUE, Second raw3.Queue;
SEIZE, Resource 1, 1:
Resource 1,1:NEXT(1495$);

1495$ DELAY: Triangular(3,4,4.8),,VA;
1494$ RELEASE: Resource 1,1;

; ; ; Model statements for module: Process 128
PROCESS module simulates the walk to the second raw of the racks.
;
1548$ QUEUE, Second raw4.Queue;
1547$ SEIZE, 2,VA:
             Resource 1,1:NEXT(1546$);

1546$ DELAY: Triangular(3,4,4.8),,VA;
1545$ RELEASE: Resource 1,1;

; ; ; Model statements for module: Process 129
PROCESS module simulates the walk to the second raw of the racks.
;
34$ ASSIGN: Second raw5.NumberIn=Second raw5.NumberIn + 1: Second raw5.WIP=Second raw5.WIP+1;
1599$ QUEUE, Second raw5.Queue;
1598$ SEIZE, 2,VA:
             Resource 1,1:NEXT(1597$);

1597$ DELAY: Triangular(3,4,4.8),,VA;
1596$ RELEASE: Resource 1,1;
1644$ ASSIGN: Second raw5.NumberOut=Second raw5.NumberOut + 1: Second raw5.WIP=Second raw5.WIP-1:NEXT(22$);

; ; ; Model statements for module: Process 130
PROCESS module simulates the walk to the third raw of the racks.
; 35$ ASSIGN: Third raw2.NumberIn=Third raw2.NumberIn + 1:
    Third raw2.WIP=Third raw2.WIP+1;
1650$ QUEUE, Third raw2.Queue;
1649$ SEIZE, 2,VA:
    Resource 1,1:NEXT(1648$);
1648$ DELAY: Triangular(7,8,8.8),,VA;
1647$ RELEASE: Resource 1,1;
1695$ ASSIGN: Third raw2.NumberOut=Third raw2.NumberOut + 1:
    Third raw2.WIP=Third raw2.WIP-1:NEXT(19$);

; ; Model statements for module: Process 131
PROCESS module simulates the walk to the third raw of the racks.
;
36$ ASSIGN: Third raw3.NumberIn=Third raw3.NumberIn + 1:
    Third raw3.WIP=Third raw3.WIP+1;
1701$ QUEUE, Third raw3.Queue;
1700$ SEIZE, 2,VA:
    Resource 1,1:NEXT(1699$);
1699$ DELAY: Triangular(7,8,8.8),,VA;
1698$ RELEASE: Resource 1,1;
1746$ ASSIGN: Third raw3.NumberOut=Third raw3.NumberOut + 1:
    Third raw3.WIP=Third raw3.WIP-1:NEXT(20$);

PROJECT, "Unnamed Project","xxx",,,No,Yes,Yes,No,No,No,No,No;

VARIABLES: Eight steps area.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Second raw2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): Picking picking list.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
Three steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Second raw1. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw3. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
If all products has been packed. NumberOut
True, CLEAR(Statistics), CATEGORY("Exclude"):
Picking picking list. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
First raw7. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
First raw2. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Second raw4. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Third raw5. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Six steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
First raw4. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
First raw7. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Six steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw1. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw6. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw3. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Three steps area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw5. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw1. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Dropping products into the box. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw3. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
First raw1. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Turning around. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw5. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw7. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Two steps area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
First raw4. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Second raw6. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Three steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
If picker 1 busy. NumberOut
False, CLEAR(Statistics), CATEGORY("Exclude"):
Four steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw1. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Dispose 1. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw2. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Third raw2. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Six steps area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Second raw3. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Third raw5. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
Four steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):
Seven steps area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):
First raw2. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
First raw4. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):
First raw6. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
First raw8. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Repeat 1. CLEAR(System), CATEGORY("User Specified/User Specified"):  
First raw6. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
First raw8. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Second raw1. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Second raw3. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Third raw4. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Two steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Seven steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Picking picking list. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw8. NumberOut, CLEAR(Statistics), CATEGORY("Exclude-Exclude"):  
First raw3. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Five steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
One step area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw2. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw4. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw5. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Two steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Five steps area. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
First raw8. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Turning around. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
If all products has been packed. NumberOut False, CLEAR(Statistics), CATEGORY("Exclude"):  
Start. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
First raw3. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Second raw5. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Eight steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw6. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Seven steps area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw1. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw1. WIP, CLEAR(System), CATEGORY("Exclude-Exclude"):  
Third raw2. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw4. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Third raw6. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw2. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw4. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
One step area. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Four steps area. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw6. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
Second raw8. NumberIn, CLEAR(Statistics), CATEGORY("Exclude"):  
First raw8. NumberOut, CLEAR(Statistics), CATEGORY("Exclude"):  
If picker 1 busy. NumberOut True, CLEAR(Statistics), CATEGORY("Exclude"):  

Turning around.

Dropping products into the box.

QUEUES:  
Biny 2.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw4.Queue,FIFO,,AUTOSTATS(Yes,,):
Third raw3.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw3.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw8.Queue,FIFO,,AUTOSTATS(Yes,,):
Two steps area.Queue,FIFO,,AUTOSTATS(Yes,,):
Biny 3.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw5.Queue,FIFO,,AUTOSTATS(Yes,,):
Eight steps area.Queue,FIFO,,AUTOSTATS(Yes,,):
Picking picking list.Queue,FIFO,,AUTOSTATS(Yes,,):
Three steps area.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw4.Queue,FIFO,,AUTOSTATS(Yes,,):
Turning around.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw4.Queue,FIFO,,AUTOSTATS(Yes,,):
Biny 4.Queue,FIFO,,AUTOSTATS(Yes,,):
Dropping products into the box.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw1.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw6.Queue,FIFO,,AUTOSTATS(Yes,,):
Third raw5.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw5.Queue,FIFO,,AUTOSTATS(Yes,,):
Hold 1.Queue,FIFO,,AUTOSTATS(Yes,,):
biny 0.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw2.Queue,FIFO,,AUTOSTATS(Yes,,):
Biny 5.Queue,FIFO,,AUTOSTATS(Yes,,):
First raw7.Queue,FIFO,,AUTOSTATS(Yes,,):
Third raw6.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw6.Queue,FIFO,,AUTOSTATS(Yes,,):
One step area.Queue,FIFO,,AUTOSTATS(Yes,,):
Third raw1.Queue,FIFO,,AUTOSTATS(Yes,,):
Second raw1.Queue,FIFO,,AUTOSTATS(Yes,,):


FAILURES:  Failure 1,Time(EXPO( 3000),EXPO( 20),): Failure 2,Time(EXPO( 3000),EXPO( 20),): Failure 3,Time(EXPO( 3000),EXPO( 20),): 

RESOURCES:  Resource 1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),FAILURE(Failure 1,Preempt),AUTOSTATS(Yes,):
COUNTERS: C1,,DATABASE(,"Count","User Specified","C1");

TALLIES: C1 total,
       "C:\Documents and Settings\Administrator\Pulpit\magisterka\A\Segment A 4 stacje\sorting solution\C1 tnow.dat",
       DATABASE(,"Expression","User Specified","C1 total");

REPLICATE, 1,,SecondsToBaseTime(54000),Yes,Yes,,,24,Seconds,No,No,,,Yes;

ENTITIES: Entity 1,Picture.Box,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
Appendix 8. The replenishment model documentation (Siman language)

Model statements for module: Create 8

CREATE module creates the entity. The entity is created with beta probability distribution.

; 2$ CREATE, 1,SecondstoBaseTime(0.0),Entity 3:SecondstoBaseTime(411 + 221 * BETA(0.156, 0.17)):NEXT(3$);

3$ ASSIGN: A1 3 from 3.NumberOut=A1 3 from 3.NumberOut + 1:NEXT(0$);

; ;
; Model statements for module: Process 1

PROCESS module simulates the activity of refilling the replenishment zone.

; 0$ ASSIGN: Refilling of replenishment zone.NumberIn=Refilling of replenishment zone.NumberIn + 1:
Refilling of replenishment zone.WIP=Refilling of replenishment zone.WIP+1;

9$ QUEUE, Refilling of replenishment zone.Queue;

8$ SEIZE, 2,VA:
Bin.filler1,1:NEXT(7$);

7$ DELAY: Triangular(1000,1800,2600),,VA;
6$ RELEASE: Bin.filler1,1;
54$ ASSIGN: Refilling of replenishment zone.NumberOut=Refilling of replenishment zone.NumberOut + 1:
Refilling of replenishment zone.WIP=Refilling of replenishment zone.WIP-1:NEXT(1$);

; ;
; Model statements for module: Dispose 1

DISPOSE module – the end of the system.

; 1$ ASSIGN: Dispose 1.NumberOut=Dispose 1.NumberOut + 1;
57$ DISPOSE: Yes;

;
CREATE module creates the entity. The entity is created with beta probability distribution.

58$ CREATE, 1, SecondstoBaseTime(0.0), Entity 2: SecondstoBaseTime(150 + 624 * BETA(0.706, 0.856)): NEXT(59$);

59$ ASSIGN: A1 2 from 3.NumberOut = A1 2 from 3.NumberOut + 1: NEXT(0$);

62$ CREATE, 1, SecondstoBaseTime(0.0), Entity 1: SecondstoBaseTime(151 + 617 * BETA(1.15, 1.22)): NEXT(63$);

63$ ASSIGN: A1 1 from 3.NumberOut = A1 1 from 3.NumberOut + 1: NEXT(0$);

66$ CREATE, 1, SecondstoBaseTime(0.0), Entity 6: SecondstoBaseTime(394 + 225 * BETA(0.137, 0.162)): NEXT(67$);

67$ ASSIGN: A2 3 from 3.NumberOut = A2 3 from 3.NumberOut + 1: NEXT(0$);
; Model statements for module: Create 13
CREATE module creates the entity. The entity is created with beta probability distribution.
;
74$ CREATE, 1,SecondstoBaseTime(0.0),Entity
4:SecondstoBaseTime(154 + 617 * BETA(1.14, 1.23)):NEXT(75$);

75$ ASSIGN: A2 1 from 3.NumberOut=A2 1 from 3.NumberOut + 1:NEXT(0$);

; Model statements for module: Create 14
CREATE module creates the entity. The entity is created with uniform probability distribution.
;
78$ CREATE, 1,SecondstoBaseTime(0.0),Entity
9:SecondstoBaseTime(UNIF(385, 763)):NEXT(79$);

79$ ASSIGN: A3 3 from 3.NumberOut=A3 3 from 3.NumberOut + 1:NEXT(0$);

; Model statements for module: Create 15
CREATE module creates the entity. The entity is created with beta probability distribution.
;
82$ CREATE, 1,SecondstoBaseTime(0.0),Entity
8:SecondstoBaseTime(147 + 626 * BETA(0.805, 0.848)):NEXT(83$);
CREATE module creates the entity. The entity is created with beta probability distribution.

CREATE, 1,SecondstoBaseTime(0.0),Entity
83$ ASSIGN: A3 2 from 3.NumberOut=A3 2 from 3.NumberOut + 1:NEXT(0$);

; cleanup
;

Model statements for module: Create 16
CREATE module creates the entity. The entity is created with beta probability distribution.

86$ CREATE, 1,SecondstoBaseTime(0.0),Entity
7:SecondstoBaseTime(154 + 621 * BETA(1.13, 1.21)):NEXT(87$);

87$ ASSIGN: A3 1 from 3.NumberOut=A3 1 from 3.NumberOut + 1:NEXT(0$);

; cleanup
;

Model statements for module: Create 18
CREATE module creates the entity. The entity is created with beta probability distribution.

90$ CREATE, 1,SecondstoBaseTime(0.0),Entity
11:SecondstoBaseTime(143 + 609 * BETA(0.91, 0.872)):NEXT(91$);

91$ ASSIGN: A4 2 from 3.NumberOut=A4 2 from 3.NumberOut + 1:NEXT(0$);

; cleanup
;

Model statements for module: Create 19
CREATE module creates the entity. The entity is created with beta probability distribution.

94$ CREATE, 1,SecondstoBaseTime(0.0),Entity
10:SecondstoBaseTime(154 + 610 * BETA(1.12, 1.17)):NEXT(95$);

95$ ASSIGN: A4 1 from 3.NumberOut=A4 1 from 3.NumberOut + 1:NEXT(0$);
CREATE module creates the entity. The entity is created with exponential probability distribution.

98$      CREATE,   1,SecondstoBaseTime(0.0),Entity
10:SecondstoBaseTime(EXPO(1674.42)):NEXT(99$);

99$      ASSIGN:    Slow zone.NumberOut=Slow zone.NumberOut + 1:NEXT(0$);

CREATE module creates the entity. The entity is created with beta probability distribution.

102$     CREATE,   1,SecondstoBaseTime(0.0),Entity
3:SecondstoBaseTime(200 + 557 * BETA(0.595, 0.778)):NEXT(103$);

103$     ASSIGN:    B1 3 from 3.NumberOut=B1 3 from 3.NumberOut + 1:NEXT(0$);

CREATE module creates the entity. The entity is created with beta probability distribution.

106$     CREATE,   1,SecondstoBaseTime(0.0),Entity
2:SecondstoBaseTime(158 + 608 * BETA(0.754, 0.899)):NEXT(107$);

107$     ASSIGN:    B1 2 from 3.NumberOut=B1 2 from 3.NumberOut + 1:NEXT(0$);

;      Model statements for module:  Create 23
CREATE module creates the entity. The entity is created with beta probability distribution.

110$ CREATE, 1,SecondstoBaseTime(0.0),Entity 1:SecondstoBaseTime(143 + 627 * BETA(1.24, 1.16)):NEXT(111$);

111$ ASSIGN: B1 1 from 3.NumberOut=B1 1 from 3.NumberOut + 1:NEXT(0$);

; ;
; Model statements for module: Create 24
CREATE module creates the entity. The entity is created with beta probability distribution.
;

114$ CREATE, 1,SecondstoBaseTime(0.0),Entity 6:SecondstoBaseTime(202 + 568 * BETA(0.567, 0.734)):NEXT(115$);

115$ ASSIGN: B2 3 from 3.NumberOut=B2 3 from 3.NumberOut + 1:NEXT(0$);

; ;
; Model statements for module: Create 25
CREATE module creates the entity. The entity is created with beta probability distribution.
;

118$ CREATE, 1,SecondstoBaseTime(0.0),Entity 5:SecondstoBaseTime(150 + 595 * BETA(0.723, 0.781)):NEXT(119$);

119$ ASSIGN: B2 2 from 3.NumberOut=B2 2 from 3.NumberOut + 1:NEXT(0$);
;
; ;
; Model statements for module: Create 26
CREATE module creates the entity. The entity is created with beta probability distribution.
;

122$ CREATE, 1,SecondstoBaseTime(0.0),Entity 4:SecondstoBaseTime(146 + 619 * BETA(1.31, 1.22)):NEXT(123$);
ASSIGN: B2 1 from 3.NumberOut = B2 1 from 3.NumberOut + 1:NEXT(0$);

PROJECT, "Unnamed Project","xxx",",,No,Yes,Yes,No,No,No,No,No;

VARIABLES: B1 2 from
3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A2 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A3 3 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A1 2 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B2 2 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
Refilling of replenishment zone.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): 
A3 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B1 3 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
Dispose 1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A1 3 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
Refilling of replenishment zone.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 
A2 2 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A4 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B1 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B2 3 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A3 2 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A2 3 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
Refilling of replenishment zone.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A1 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
Slow zone.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B2 1 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A4 2 from 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");

QUEUES: Refilling of replenishment zone.Queue,FIFO,,AUTOSTATS(Yes,);

PICTURES: 
Picture.Airplane:
Picture.Green Ball:
Picture.Blue Page:
Picture.Telephone:
Picture.Blue Ball:
Picture.Yellow Page:
Picture.EMail:
Picture.Yellow Ball:
Picture.Bike:
Picture.Report:
RESOURCES:
Bin.filler1,Capacity(84),,COST(0.0,0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS (Yes,);

REPLICATE, 1,,SecondsToBaseTime(540000),Yes,Yes,,,24,Seconds,No,No,,,Yes;

ENTITIES:  Entity 1,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 2,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 3,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 4,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 5,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 6,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 7,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 8,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 9,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 10,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
 Entity 11,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,);
Model statements for module: Create 1

CREATE module creates the entity. The entity is created with exponential probability distribution.

46$     CREATE,
        1, SecondstoBaseTime(0.0), Order: SecondstoBaseTime(EXPO(5.8)): NEXT(47$);

47$     ASSIGN:        Start point. NumberOut = Start point. NumberOut + 1: NEXT(45$);

; ; ;
; Model statements for module: Decide 10

DECIDE module checks which station has the smallest number of orders in the queue. Station considered from fast moving products zone: 1, 2, 3; 4.

45$     BRANCH, 1:
        If,
        MIN(NQ(A1 station.Queue), NQ(A2 station.Queue), NQ(A3 station.Queue),
        NQ(A4 station.Queue)) == NQ(A1 station.Queue),
        1$, Yes:
        If,
        MIN(NQ(A1 station.Queue), NQ(A2 station.Queue), NQ(A3 station.Queue),
        NQ(A4 station.Queue)) == NQ(A2 station.Queue),
        4$, Yes:
        If,
        MIN(NQ(A1 station.Queue), NQ(A2 station.Queue), NQ(A3 station.Queue),
        NQ(A4 station.Queue)) == NQ(A3 station.Queue),
        5$, Yes:
        Else, 6$, Yes;

; ; ;
; Model statements for module: Process 7

PROCESS module simulates process of picking the products on station 4 in fast moving products zone.

6$     ASSIGN:        A4 station. NumberIn = A4 station. NumberIn + 1:
        A4 station. WIP = A4 station. WIP + 1;

55$     QUEUE, A4 station. Queue;

54$     SEIZE, 2, VA:
        picker4, 1: NEXT(53$);
DELAY: 89 * BETA(2.55, 39.7), VA;
RELEASE: picker4,1;
ASSIGN: A4 station.NumberOut = A4 station.NumberOut + 1:
        A4 station.WIP = A4 station.WIP - 1: NEXT(17$);

Model statements for module: Assign 11
ASSIGN module assigns to entity variable A4 with value of one, which confirms that
order (entity) was already served by station 4 in fast moving products zone.

ASSIGN: A4 = 1: NEXT(23$);

Model statements for module: Decide 19
DECIDE module checks whether entity (order) had been already served by the station
1 in fast moving products zone.

BRANCH, 1:
        If A1 == 1, 103$, Yes:
        Else, 104$, Yes;
ASSIGN: If A1 served.NumberOut True = If A1 served.NumberOut True + 1: NEXT(18$);
ASSIGN: If A1 served.NumberOut False = If A1 served.NumberOut False + 1: NEXT(24$);

Model statements for module: Decide 14
DECIDE module checks whether entity (order) had been already served by the station
2 in fast moving products zone.

BRANCH, 1:
        If A2 == 1, 105$, Yes:
        Else, 106$, Yes;
ASSIGN: If A2 served2.NumberOut True = If A2 served2.NumberOut True + 1: NEXT(19$);
ASSIGN: If A2 served2.NumberOut False = If A2 served2.NumberOut False + 1: NEXT(21$);
DECIDE module checks whether entity (order) had been already served by the station 3 in fast moving products zone.

19$ BRANCH, 1:
    If, A3==1, 107$, Yes:
    Else, 108$, Yes;
107$ ASSIGN: If A3 served3.NumberOut True=If A3 served3.NumberOut True + 1:NEXT(20$);

108$ ASSIGN: If A3 served3.NumberOut False=If A3 served3.NumberOut False + 1:NEXT(27$);

DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

20$ BRANCH, 1:
    If, A4==1, 109$, Yes:
    Else, 110$, Yes;
109$ ASSIGN: If A4 served7.NumberOut True=If A4 served7.NumberOut True + 1:NEXT(10$);

110$ ASSIGN: If A4 served7.NumberOut False=If A4 served7.NumberOut False + 1:NEXT(6$);

DECIDE module checks whether entity (order) had been already served medium moving products zone.

10$ BRANCH, 1:
    With, 98/100, 111$, Yes:
    Else, 112$, Yes;
111$ ASSIGN: B zone visited.NumberOut True=B zone visited.NumberOut True + 1:NEXT(39$);
ASSIGN: B zone visited.NumberOut False=B zone visited.NumberOut False + 1:NEXT(9$);

; ;
; ; Model statements for module: Decide 78
DECIDE module checks which station in medium moving products zone has the smallest number of orders in the queue.
;
39$ BRANCH, 1:
    If,MIN(NQ(B1 station.Queue),NQ(B2 station.Queue)) == NQ(B1 station.Queue),2$,Yes:
    Else,7$,Yes;

; ;
; ; Model statements for module: Process 8
PROCESS module simulates process of picking the products on station 2 in medium moving products zone.
;
7$ ASSIGN: B2 station.NumberIn=B2 station.NumberIn + 1:
    B2 station.WIP=B2 station.WIP+1;
118$ QUEUE, B2 station.Queue;
117$ SEIZE, 2,VA:
    picker6,1:NEXT(116$);
116$ DELAY: 55 * BETA(0.457, 5.36),,VA;
115$ RELEASE: picker6,1;
163$ ASSIGN: B2 station.NumberOut=B2 station.NumberOut + 1:
    B2 station.WIP=B2 station.WIP-1:NEXT(41$);

; ;
; ; Model statements for module: Assign 21
ASSIGN module assigns to entity variable B2 with value of one, which confirms that order (entity) was already served by station 2 in medium moving products zone.
;
41$ ASSIGN: B2=1:NEXT(43$);

; ;
; ; Model statements for module: Decide 84
DECIDE module checks whether entity (order) had been already served by the station 1 in medium moving products zone.

: 
43$ BRANCH, 1: 
   If, B1==1, 166$, Yes:
   Else, 167$, Yes;
166$ ASSIGN: If B1 served.NumberOut True=If B1 served.NumberOut True + 1:NEXT(42$);

167$ ASSIGN: If B1 served.NumberOut False=If B1 served.NumberOut False + 1:NEXT(44$);

;
;
; Model statements for module: Decide 79

DECIDE module checks whether entity (order) had been already served by the station 2 in medium moving products zone.

: 
42$ BRANCH, 1: 
   If, B2==1, 168$, Yes:
   Else, 169$, Yes;
168$ ASSIGN: If B2 served.NumberOut True=If B2 served.NumberOut True + 1:NEXT(9$);

169$ ASSIGN: If B2 served.NumberOut False=If B2 served.NumberOut False + 1:NEXT(7$);

;
;
; Model statements for module: Decide 4

DECIDE module determines the orders which does not have anything to be picked within slow moving products zone.

: 
9$ BRANCH, 1: 
   With, 24/100, 170$, Yes:
   Else, 171$, Yes;
170$ ASSIGN: C zone visited.NumberOut True=C zone visited.NumberOut True + 1:NEXT(3$);

171$ ASSIGN: C zone visited.NumberOut False=C zone visited.NumberOut False + 1:NEXT(0$);
PROCESS module simulates process of picking the products on station 1 in slow moving products zone.

ASSIGN:  C1 station.NumberIn=C1 station.NumberIn + 1:
          C1 station.WIP=C1 station.WIP+1;
QUEUE,  C1 station.Queue;
SEIZE,  2,VA:
          picker9,1:NEXT(173$);
DELAY:  4 + GAMM(2.54, 3.76),,VA;
RELEASE: picker9,1;
ASSIGN:  C1 station.NumberOut=C1 station.NumberOut + 1:
          C1 station.WIP=C1 station.WIP-1:NEXT(0$);

Model statements for module: Dispose 1

ASSIGN:  End point.NumberOut=End point.NumberOut + 1;
DISPOSE: Yes;

Model statements for module: Decide 85
DECIDE module checks whether entity (order) had been already served by the station 2 in medium moving products zone.

BRANCH,  1:
          If,B2==1,224$,Yes:
          Else,225$,Yes;
ASSIGN:  If B2 served2.NumberOut True=If B2 served2.NumberOut True + 1:NEXT(2$);
ASSIGN:  If B2 served2.NumberOut False=If B2 served2.NumberOut False + 1:NEXT(39$);

Model statements for module: Process 3
PROCESS module simulates process of picking the products on station 1 in medium moving products zone.
ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.

ASSIGN module assigns to entity variable B1 with value of one, which confirms that order (entity) was already served by station 1 in medium moving products zone.
Model statements for module: Assign 10
ASSIGN module assigns to entity variable A3 with value of one, which confirms that order (entity) was already served by station 3 in fast moving products zone.

Model statements for module: Decide 33
DECIDE module checks which station has the smallest number of orders in the queue. Station considered: 3, 4 in fast moving products zone.

Model statements for module: Decide 17
DECIDE module checks whether entity (order) had been already served by the station 3 in fast moving products zone.
DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

\[
\begin{align*}
32\$ & \quad \text{BRANCH, 1:} \\
& \quad \text{If } A_4 == 1, 334\$, Yes; \\
& \quad \text{Else, 335\$, Yes;}
\end{align*}
\]

\[
\begin{align*}
334\$ & \quad \text{ASSIGN: If } A_4 \text{ served}_6.\text{NumberOut True} = \text{If } A_4 \text{ served}_6.\text{NumberOut True} + 1: \text{NEXT(4$)}; \\
335\$ & \quad \text{ASSIGN: If } A_4 \text{ served}_6.\text{NumberOut False} = \text{If } A_4 \text{ served}_6.\text{NumberOut False} + 1: \text{NEXT(36$)};
\end{align*}
\]

PROCESS module simulates process of picking the products on station 2 in fast moving products zone.

\[
\begin{align*}
4\$ & \quad \text{ASSIGN: A}_2 \text{ station.NumberIn} = \text{A}_2 \text{ station.NumberIn} + 1: \\
& \quad \text{A}_2 \text{ station.WIP} = \text{A}_2 \text{ station.WIP} + 1; \\
339\$ & \quad \text{QUEUE, A}_2 \text{ station.Queue;} \\
338\$ & \quad \text{SEIZE, 2, VA: picker2,1: NEXT(337$)};
\end{align*}
\]

\[
\begin{align*}
337\$ & \quad \text{DELAY: GAMM(2.42, 2.29),, VA;} \\
336\$ & \quad \text{RELEASE: picker2,1;} \\
384\$ & \quad \text{ASSIGN: A}_2 \text{ station.NumberOut} = \text{A}_2 \text{ station.NumberOut} + 1: \\
& \quad \text{A}_2 \text{ station.WIP} = \text{A}_2 \text{ station.WIP} - 1: \text{NEXT(15$)};
\end{align*}
\]

ASSIGN module assigns to entity variable A2 with value of one, which confirms that order (entity) was already served by station 2 in fast moving products zone.

\[
\begin{align*}
15\$ & \quad \text{ASSIGN: A2 = 1: NEXT(23$)};
\end{align*}
\]

Model statements for module: Decide 32
DECADE module checks which station from considered has the smallest number of orders in the queue. Stations considered: 2, 4 in fast moving products zone.

\[
\begin{align*}
36$ & \quad \text{BRANCH, 1:} \\
& \quad \text{If, } \min(\text{NQ(A2 station.Queue), NQ(A4 station.Queue)}) == \text{NQ(A2 station.Queue)}, 1$, Yes:
& \quad \text{Else, 6$, Yes;}
\end{align*}
\]

\[
\begin{align*}
1$ & \quad \text{ASSIGN: A1 station.NumberIn=A1 station.NumberIn + 1:} \\
& \quad \text{A1 station.WIP=A1 station.WIP+1;}
392$ & \quad \text{QUEUE, A1 station.Queue;}
391$ & \quad \text{SEIZE, 2, VA:} \\
& \quad \text{picker1,1:NEXT(390$);}
390$ & \quad \text{DELAY: GAMM(2.41, 2.29), VA;}
389$ & \quad \text{RELEASE: picker1,1;}
437$ & \quad \text{ASSIGN: A1 station.NumberOut=A1 station.NumberOut + 1:} \\
& \quad \text{A1 station.WIP=A1 station.WIP-1:NEXT(14$;)
\end{align*}
\]

Model statements for module: Assign 8

ASSIGN module assigns to entity variable A1 with value of one, which confirms that order (entity) was already served by station 1 in fast moving products zone.

\[
\begin{align*}
14$ & \quad \text{ASSIGN: A1=1:NEXT(23$;)
\end{align*}
\]

Model statements for module: Decide 18

DECADE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

\[
\begin{align*}
22$ & \quad \text{BRANCH, 1:} \\
& \quad \text{If, A4==1, 440$, Yes:} \\
& \quad \text{Else, 441$, Yes;}
440$ & \quad \text{ASSIGN: If A4 served8.NumberOut True=If A4 served8.NumberOut True + 1:NEXT(35$;)
\end{align*}
\]
ASSIGN: If A4 served.NumberOut False=If A4 served.NumberOut False + 1:NEXT(8$);

; ; Model statements for module: Decide 31
DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 2, 3 in fast moving products zone.
; 35$ BRANCH, 1:
    If,\(\text{MIN}(\text{NQ(A2 station.Queue)},\text{NQ(A3 station.Queue)}) = \text{NQ(A2 station.Queue)}\),4$,Yes:
    Else,5$,Yes;

; ; Model statements for module: Decide 1
DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 2, 3, 4 in fast moving products zone.
; 8$ BRANCH, 1:
    If,\(\text{MIN}(\text{NQ(A2 station.Queue)},\text{NQ(A3 station.Queue)},\text{NQ(A4 station.Queue)}) = \text{NQ(A2 station.Queue)}\),4$,,
        Yes:
        If,\(\text{MIN}(\text{NQ(A2 station.Queue)},\text{NQ(A3 station.Queue)},\text{NQ(A4 station.Queue)}) = \text{NQ(A3 station.Queue)}\),5$,
            Yes:
            Else,6$,Yes;

; ; Model statements for module: Decide 20
DECIDE module checks whether entity (order) had been already served by the station 2 in fast moving products zone.
; 24$ BRANCH, 1:
    If,A2==1,446$,Yes:
    Else,447$,Yes;
446$ ASSIGN: If A2 served.NumberOut True=If A2 served.NumberOut True + 1:NEXT(28$);
447$ ASSIGN: If A2 served.NumberOut False=If A2 served.NumberOut False + 1:NEXT(25$);
DECIDE module checks whether entity (order) had been already served by the station 3 in fast moving products zone.

28$  BRANCH,  1:
      If,A3==1,448$,Yes:
      Else,449$,Yes;
448$  ASSIGN:  If A3 served2.NumberOut True=If A3 served2.NumberOut True + 1:NEXT(29$);
449$  ASSIGN:  If A3 served2.NumberOut False=If A3 served2.NumberOut False + 1:NEXT(31$);

DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

29$  BRANCH,  1:
      If,A4==1,450$,Yes:
      Else,451$,Yes;
450$  ASSIGN:  If A4 served3.NumberOut True=If A4 served3.NumberOut True + 1:NEXT(1$);
451$  ASSIGN:  If A4 served3.NumberOut False=If A4 served3.NumberOut False + 1:NEXT(38$);

DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 4 in fast moving products zone.

38$  BRANCH,  1:
      If,MIN(NQ(A1 station.Queue),NQ(A4 station.Queue)) == NQ(A1 station.Queue),1$,Yes:
      Else,6$,Yes;

; Model statements for module: Decide 27
DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

; 31$ BRANCH, 1:
   If, A4==1,454$, Yes:
   Else, 455$, Yes;
454$ ASSIGN: If A4 served5.NumberOut True=If A4 served5.NumberOut True + 1:NEXT(34$);
455$ ASSIGN: If A4 served5.NumberOut False=If A4 served5.NumberOut False + 1:NEXT(11$);

; ; Model statements for module: Decide 30
DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 3 in fast moving products zone.

; 34$ BRANCH, 1:
   If, MIN(NQ(A1 station.Queue), NQ(A3 station.Queue)) == NQ(A1 station.Queue), 1$, Yes:
   Else, 5$, Yes;

; ; Model statements for module: Decide 11
DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 3, 4 in fast moving products zone.

; 11$ BRANCH, 1:
   If, MIN(NQ(A1 station.Queue), NQ(A3 station.Queue), NQ(A4 station.Queue)) == NQ(A1 station.Queue), 1$, Yes:
   If, MIN(NQ(A1 station.Queue), NQ(A3 station.Queue), NQ(A4 station.Queue)) == NQ(A3 station.Queue), 5$, Yes:
   Else, 6$, Yes;

; ; Model statements for module: Decide 21
DECIDE module checks whether entity (order) had been already served by the station 3 in fast moving products zone.
DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.

DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 2 in fast moving products zone.

DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 2, 4 in fast moving products zone.
Yes:
If,MIN(NQ(A2 station.Queue),NQ(A1 station.Queue),NQ(A4 station.Queue)) == NQ(A2 station.Queue),4$,
Yes:
Else,6$,Yes;

; ;
; Model statements for module: Decide 22
DECIDE module checks whether entity (order) had been already served by the station 4 in fast moving products zone.
;
26$ BRANCH, 1:
If,A4==1,468$,Yes:
Else,469$,Yes;
468$ ASSIGN: If A4 served.NumberOut True=If A4 served.NumberOut True + 1:NEXT(13$);
469$ ASSIGN: If A4 served.NumberOut False=If A4 served.NumberOut False + 1:NEXT(45$);

; ;
; Model statements for module: Decide 13
DECIDE module checks which station from considered has the smallest number of orders in the queue. Station considered: 1, 2, 3 in fast moving products zone.
;
13$ BRANCH, 1:
If,MIN(NQ(A2 station.Queue),NQ(A3 station.Queue),NQ(A1 station.Queue)) == NQ(A1 station.Queue),1$,
Yes:
If,MIN(NQ(A2 station.Queue),NQ(A3 station.Queue),NQ(A1 station.Queue)) == NQ(A2 station.Queue),4$,
Yes:
Else,5$,Yes;

PROJECT, "Unnamed Project","xxx",,,No,Yes,Yes,No,No,No,No,No;

ATTRIBUTES:  A1:
A2:
A3:
A4:
VARIABLES: A1 station.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
A3 station.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served5.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
B1: 
B2: 

If A4 served5.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
If A3 served.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
A4 station.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
End point.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served2.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
B2 station.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
B zone visited.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
If A3 served3.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
C zone visited.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
True,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
A2 station.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served6.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
False,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served6.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served2.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
If A3 served4.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served7.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
B1 station.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 
C1 station.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served2.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"): 
If A4 served3.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude-Exclude"): 
If A4 served7.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
B1 station.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 
C1 station.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"): 
If A3 served2.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"): 
If B2 served2.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude-Exclude"): 
A1 station.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 
B2 station.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 
A1 station.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"): 

230

QUESTES: A3 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
B2 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
C1 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
A4 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
A1.Queue,FIFO,.,AUTOSTATS(Yes,.):
A1 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
A2 station.Queue,FIFO,.,AUTOSTATS(Yes,.):
B1 station.Queue,FIFO,.,AUTOSTATS(Yes,.):

PICTURES: Picture.Airplane:
RESOURCES:
picker1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker2,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker3,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker4,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker5,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker6,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
picker9,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes ,):
REPLICATE, 10,,SecondsToBaseTime(54000),Yes,Yes,,,24,Seconds,No,No,,,Yes;

ENTITIES: Order,Pi