A study into optimizing the layout of production facilities by using computer simulation to optimize interdepartmental flow. Tested PaperFoam's three existing production locations.

A simulation based layout optimization study for production facilities

T.S. Teoh

PaperFoam®

UNIVERSITY OF TWENTE.

Preface

I would like to thank PaperFoam for the unique opportunity they provided me. Especially sending me to both foreign production locations in the US and Malaysia was an honor. I would especially like to thank Roel Groenveld and Martin van Zandwijk. Martin, thank you for the many coffees on the long drive from the island to the factory! I know the research took way longer than expected (and hoped), but I hope you can still be proud of the results.

From the University of Twente I would like to thank my supervisors Martijn Mes and Marco Schutten. Thank you for your many feedback rounds and immeasurable patience.

Management Summary

PaperFoam is a packaging manufacturing company based in Barneveld, The Netherlands. They have three production facilities, one in Barneveld, The Netherlands, one in Leland, North Carolina, US and one in Penang, Malaysia. They make packaging material based on paper fiber and potato starch, resulting in a biodegradable product.

The current locations have grown naturally and the increasing demand for sustainable packaging yielded the demand for another production facility. Both aspects gave the need for a layout optimization study.

This resulted in our main research question:

"How can current layout optimization models be improved and adapted to a new generic model to optimize layouts for production facilities?"

To answer this question we first performed a literature study to create scientific background and identify possible improvements. We then analyzed all three existing locations and identified the main waste in interdepartmental flow, namely the movement from the operators transporting vessels with material between the mixing machines and the production lines.

We started our research with a literature study where we discussed several well-known layout optimization models. We found that all models either need a lot of input data, which usually is not available, or use straight lines to calculate the walking time or distance between two objects. We found the solution for this in computer simulation. This solution can cope with limited input data and uses realistic distances between attributes. We introduced several steps needed to validate and verification computer simulations and have used these steps to later validate our simulation model.

We then build a generic model based on the facility layout problem. In this model we defined two layout creation methods, namely randomly placing attributes and placing them along all four outer walls. We used local search with simulated annealing to further search for layout optimizations within the created layouts. To investigate the effect local search had on a generated layout we ran the same layout with and without local search and found up to a 11% reduction in total walk time. The simulation model was programmed into Tecnomatix Plant Simulation 13. After the simulation model was made we validated the outcome with a given historical production schedule. We also used this samples from this production schedule to run the experiments.

To see the effect of using a computer simulation model on the estimated walking distance and time we used the original layout of the Dutch facility. We ran the simulation (with the 6 replications) and found the number of walks operators had performed to each machine. We calculated the distance from the center of the mixers to all the machines using a straight path (crossing obstacles). We then estimated the required walking time for the operators using the same number of walks to each machine. We found that on average the distance using a straight path was 18.5% lower, with an extreme of 50% lower. The total walking time was 20% lower, thus resulting in an underestimation of the required walking time.

We then ran the simulation optimization model for all three. We ran the simulation for 15 days and had 6 replications per tested layout. The original layout of the Dutch facility had a total walking time of 6 hours, 47 minutes and 26 seconds. The best-found layout had a total walking time of 4 hours, 39 minutes and 53 seconds. A total reduction in walking time of 31%, or 2 hours, 7 minutes and 33 seconds. The original layout of the American facility had a total walking time of 9 hours, 58 minutes and 8 seconds. The best-found time had a total walking time of 5 hours, 13 minutes and 5 seconds. A total walking time reduction of 4 hours, 45 minutes and 3 seconds is achieved. This is a total reduction of 48%. Lastly, the Malaysian facility originally had a walking time of 10 days, 7 hours, 41 minutes and 33 seconds. The best-found layout had a total walking time of 9 days, 15 hours, 59 minutes and 50 seconds, a reduction of 15 hours, 41 minutes and 43 seconds. Although this is a small percentage reduction, namely 7%, it still is a decent absolute reduction.

So to conclude our research we can state that the current layout optimization methods can be improved by using more realistic walking distances. This can be achieved by using computer simulation, since this takes obstacles that have to be avoided into account. With the computer simulation model that we have created we could easily generate better layouts for all three existing locations of PaperFoam, reaching up to a 48% reduction in total walking time.

Table of contents

1 Introduction
In Section 1.1, we give a brief introduction into the company PaperFoam. Section 1.2 explains the reason for this research and Section 1.3 introduces the proposed research plan. Section 1.4 explains the outline of the report.

1.1 Company background
This section will give a short introduction into the company PaperFoam.

PaperFoam is a producer of green packaging material. They mainly produce the packaging inserts that hold the products into place. This packaging material consist of a mixture of industrial starch, natural fibers, water and their patented premix (see Figure 1) and is produced using injection molding. The carbon emissions are 90% lower compared to their plastic counterparts. The clientele of PaperFoam mainly consists of consumer electronics manufacturers like Valve, Philips and Plantronics, but their products are also used in other industries, like to pack medical devices, dry foods or cosmetics. Rituals is one of the customers in the latter industry. The finished product is made using a blow molding procedure where the batter is pumped from the vessels into preheated molds. Depending on the size of the finished product this mold can have 1 up to 12 cavities. While the batter touches the hot mold, the water in the batter starts to evaporate, making the mixture foam. This results in a lightweight product. After a predetermined time the mold will open, dropping the dried up products out of the machine. The closing of the mold, injecting of the batter, cooking time, opening of the mold and the finished products dropping out will be called one stroke.

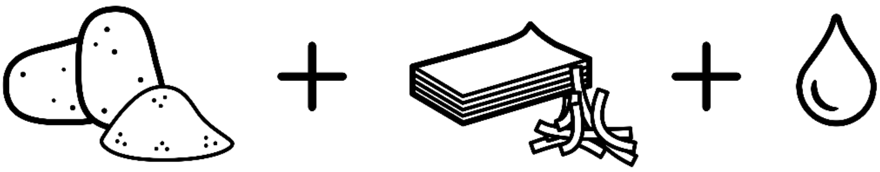


Figure 1: Ingredients for the batter of PaperFoam

The headquarter of PaperFoam is located in Barneveld, The Netherlands. They have production facilities in Barneveld, The Netherlands, in Leland, USA and in Penang, Malaysia. They are in the process of opening a fourth production facility in Poland. They also have an experience center in San Francisco, USA.

1.2 Project background
This section will give some background information behind our research.

The demand for sustainable packing solutions is growing rapidly and PaperFoam predicts that they need to open more facilities to cope with the demand. Furthermore, they want to get insight into the costs of the material flow in their three existing production facilities. After some observations and talking to the workers we found out that mainly the interdepartmental flow of material and raw material could be optimized. PaperFoam wants to find out what the interdepartmental transportation cost at the current locations are, and how to minimize the operational costs of future facilities. In our research we will focus on the layout of the production part of the company. Since these are existing facilities, PaperFoam is limited in their freedom to change the layout, this is a so called "brownfield" factory redesign.

In addition, PaperFoam plans to open a few more production facilities in the coming years. The locations are yet to be determined, but a better understanding of resource requirements will help with that decision. This is a so called "greenfield" factory design, which gives more freedom in the layout design. But PaperFoam prefers to have some sort of standardization in their layout designs, to make it easier and more effective to manage and maintain.

PaperFoam also wants to automate parts of their production process. In order to see what steps in the process are suitable, a better understanding of the flow, and especially its time requirement is desired.

1.3 Research plan

This section will explain what problem we try to solve. It introduces the research questions we use to formulate a solution and it will introduce the approach we will use to reach the solution.

Problem Definition

In Barneveld, PaperFoam opened a second production hall early 2018. This resulted in twice the production capacity, but this did not necessarily result in an improvement in the material flow through the facility. They have noticed that the total number of man-hours per finished product is higher than at their Leland facility, which has roughly the same salary per hour per function. The personnel cost is around 30-35% of the price of the total finished product. The Malaysian facility consist of two separate buildings. Both buildings also have two floors, where the top floors are production areas and in one of the two buildings the bottom floor has the mixing area, this means that raw and finished material has to move between the buildings and floors. To reduce the personnel cost PaperFoam wants to investigate if an improvement in the material flow would yield lower costs. Furthermore they want to be able to easily create new layouts for new facilities. We have observed that most of the transportation is the movement of the transportation vessels containing the batter for the production machines, so the study will focus on minimizing the required movement.

Research questions

To structure the research the following research question was formed:

"How can current layout optimization models be improved and adapted to a new generic model to optimize layouts for production facilities?"

To find an answer to this research question, it is divided it into 6 research sub questions. The first research question will be a literature study looking into the possible solutions to the layout problem and material flow design that already has been created and what their possible shortcomings are. After the existing literature is analyzed, we will analyze the current situation in all three existing. The third research question will be used to create a model that can quickly generate layout alternatives for both existing (brownfield) and new production (greenfield) locations. The fourth research question investigate how we can translate this theoretical model into a simulation model. The fifth research question will answer how we can validate that the created simulation model correctly represents the reality. In the sixth research question will analyze the layouts of the three existing production facilities analyzed in research question two, using the model created in research question three.

We have the following research sub questions:

1. **What can be found in the current literature about layout optimization models?**
   What models for the layout problem and material flow design are known in the current literature? What are the drawbacks of these models? What new techniques can we use to improve the existing models?
2. **What is the current situations at the three existing PaperFoam production facilities?**
   What departments do the locations have? What is the layout of the facility? What and how much is the interdepartmental material flow? Where are the pick-up and drop-off points for the interdepartmental material flow? Which resources, especially personnel, are needed to produce the final product? What is the problem with the current layout that makes it less efficient?
   a. For the first location: Barneveld, The Netherlands
   b. For the second location: Leland, North Carolina, United States
   c. For the third location: Penang, Malaysia
3. **How can we develop a general model to create better layouts for production facilities?**
   What model can we develop, using the improvements found in the second research sub question, to quickly generate layout alternatives and be able to rate them, for either existing production locations (brownfield) or new locations (greenfield)?
4. **How can we implement the model using computer software?**
   How do we translate the theoretical model to a coded simulation model? How many replications are needed? What is the required run time? How can the simulation model be validated?
5. **Are the results from the computer simulation statically comparable to reality?**
   Can we subjectively validate the simulation model? Can we objectively validate the simulation model?
6. **How do the generated optimized layouts perform compared to the existing situations?**
   What is the main difference between the original layout and the proposed improved layout? Where does the saving come from and how much is saved compared to the original layout? How much man-hours and other resources are needed to cope with all interdepartmental logistics?

Research approach

We will start with a literature study to investigate what already has been done and where the gaps within the existing literature are. Our research then continues with closely observing the three current production facilities of PaperFoam. The observations will be performed by following a person from every department and record every action they perform. This ensures that we truly understand all the processes that take place within the production facility. Furthermore, we will analyze data given by PaperFoam. From these observations we will answer the second research sub question. This will give a clear picture of the current situation at all three locations and where the room for improvement is.

We will then create a model for creating alternative layouts for existing and new locations. This model will include the already existing techniques and a solution for the identified gaps in the research. One of the disadvantages of the current literature is that it can only cope with one input or output point. Furthermore, the existing models use either Euclidean or Rectilinear (also called Manhattan) distances, both are not exactly precise. Since in our study the production lines have more input and output points and the exact distance is important, we will use computer simulation. We will also use Simulation Optimization to be able to more realistically score the different layouts and find an efficient one.

Finally, we will use the created model to analyze three existing production locations of PaperFoam and to create a new layout for a new facility.

Research objectives and scope

The objective of this research is twofold. First, to get a precise understanding of the employee costs at the current three production facilities. Secondly, we will create a model that PaperFoam, and other production facilities, can use to quickly and easily generate new efficient layout design given some constraints.

Since we mainly focus on the interdepartmental logistics, we assume an infinite supply of raw material and an infinite demand for finished product. In other words, we are not going to optimize the ordering and delivery of incoming materials, we do take the transportation from the unloading bay to the location where the raw material is needed into account. We also do not focus on generating optimal production schedules, we will be using the actual schedules that the production manager also uses. We will also only focus on the material flow, not the information flow, we assume that all the employees have all the information required for them to do their job.

1.4 Outline of the report

This section will explain the structure the report will have.

The report has the following structure. In Chapter 2 we describe the performed literature study and its conclusions. Chapter 3 describes the current situation of the three production facilities of PaperFoam. This chapter explains the flow through the facility, the differences between the three locations and where there is room for improvements. Chapter 4 explains the creation of the theoretical model that production facilities can use to develop alternative layouts and determine which one is efficient. In Chapter 5 we use the developed model to create a simulation model. This simulation model is used to generate alternative layouts for PaperFoam's existing production facilities. In Chapter 6 the created simulation model will be validated. In Chapter 7 we analyze the performance of the developed model against the original situations. Chapter 8 will give the conclusion from our research and we will give a discussion and recommendations for further research.

2 Literature study

To have a clear understanding of what already has been researched and where there are still things to be investigated, we do a literature study into all relevant fields for our research. Section 2.1 will be about the Facility Layout Problem. Section 2.2 will be about material flow and Section 2.3 about simulation. Section 2.3 explains techniques to evaluate layouts. Section 2.5 gives a conclusion to the literature study.

2.1 Facility Layout Problem

This section will introduce the Facility Layout Problem.

Tompkins et al. (2010) state that 20 to 50% of the manufacturing costs are due to the handling of parts and then a good arrangement of handling devices might reduce those costs to 10 to 30%. Drira, Pierreval and Hajri-Babouj (2007) state that a facility layout is an arrangement of everything needed for production of goods or delivery of service. Layout problems can be split into static and dynamic layout problems. Researchers do not agree on a common and exact definition of layout problems. The most encountered definition for layout problems is by Koopmans and Beckmann (1957) and goes as follows: A common industrial problem in which the objective is to configure facilities, so as to minimize the cost of transporting materials between them.

Drira, Pierreval and Hajri-Babouj (2007) state that mostly older literature considers layouts as being static; they assume that the key data about the facility and what it is intended to produce will remain constant over a long period of time. More recently, the idea of dynamic layout problems have been introduced by several researchers (Balakrishnan & Cheng, 1998) (Braglia, Zanoni, & Zavanella, 2003). Dynamic layout problems take into account possible changes in the material handling flow. Drira, Pierreval and Hajri-Babouj (2007) further state that a layout plan for the dynamic layout problem consists of series of layouts, each layout being associated with a period. Baykasoglu and Gindy (2001) states that rearrangement costs have to be considered when facilities or machines need to be moved from one location to another.

Chhajed, Montreuil and Lowe (1992) state that one way of solving a facility layout planning problem is to use a component approach. They divide the problem in four components, namely a) block design, b) input/output station location, c) material flow network design and d) aisle netting. Depending on how the problem is formulated, it has to be approached discrete or continuous. In the literature, the most common way of solving a discrete layout problem is by using *Quadratic Assignment Problems* (QAP) and *Mixed Integer Programming* (MIP). Figure 2 shows a discrete layout. Fruggiero, Lambiase and Negri (2006) address this problem as QAP. Here the plant is divided into rectangular blocks with the same shape and area. Each block is then assigned to a facility. Figure 2 also shows a continuous layout. The block design can be divided in two different analytical approaches, namely 1) the *quadratic assignment formulation* (Koopmans & Beckmann, 1957) and 2) the *graph-theoretic approach* (Foulds, 1983). Both approaches only derive block plans. Operational details like circulation regions, aisle structures and the location of the input and output station are generally not modeled. Several researchers (O'Brien & Abdul Barr, 1980) recognize that considering aisle travel in major layout design provides significant potential for improvement in flow travel and space devoted to the aisles.

Das (1993) formulates this problem as a MIP. All the facilities are placed anywhere within the planar site and must not overlap each other. Tompkins, White, Bozer and Tanchoco (2010) define the problem as follows.

B_x be the building length (measured along the x-coordinate)
B_y be the building width (measured along the y-coordinate)
A_i be the area of department i
L_i^l be the lower limit on the length of department i
L_i^u be the upper limit on the length of department i
W_i^l be the lower limit on the width of department i
W_i^u be the upper limit on the width of department i
M be a large number

With the next decision variables, let:

α_i be the x-coordinate of the centroid of department i
β_i be the y-coordinate of the centroid of department i
f_{ij} be the interdepartmental flow from department i to j
c_{ij} be the cost of moving a unit of material from department i to j
x_i' be the x-coordinate of the left (or west) side of department i
x_i'' be the x-coordinate of the right (or east) side of department i
y_i' be the y-coordinate of the top (or north) side of department i
y_i'' be the y-coordinate of the bottom (or south) side of department i
z_{ij}^x be 1 if department i is strictly to the east of department j, and 0 otherwise
z_{ij}^y be 1 if department i is strictly to the north of department j, and 0 otherwise

$$z = min \sum_i \sum_j f_{ij} * c_{ij} * \left(\left|\alpha_i - \alpha_j\right| + \left|\beta_i - \beta_j\right|\right) \qquad (2.1)$$

Subject to:

$L_i^l \leq (x_i'' - x_i') \leq L_i^u$	for all i	(2.2)
$W_i^l \leq (y_i'' - y_i') \leq W_i^u$	for all i	(2.3)
$(x_i'' - x_i') * (y_i'' - y_i') = A_i$	for all i	(2.4)
$0 \leq x_i' \leq x_i'' \leq B_x$	for all i	(2.5)
$0 \leq y_i' \leq y_i'' \leq B_y$	for all i	(2.6)
$\alpha_i = 0.5 * x_i' + 0.5 * x_i''$	for all i	(2.7)
$\beta_i = 0.5 * y_i' + 0.5 * y_i''$	for all i	(2.8)
$x_j'' \leq x_i' + M * \left(1 - z_{ij}^x\right)$	for all i and j, i ≠ j	(2.9)
$y_j'' \leq y_i' + M * \left(1 - z_{ij}^y\right)$	for all i and j, i ≠ j	(2.10)
$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1$	for all i and j, i < j	(2.11)
$\alpha_i, \beta_i \geq 0$	for all i	(2.12)
$x_i', x_i'', y_i', y_i'' \geq 0$	for all i	(2.13)
z_{ij}^x, z_{ij}^y $0/1$ $integer$	for all i and j, i ≠ j	(2.14)

Constraints 2.2 and 2.3 ensure that the length and width of each department are within the specified bounds. The area requirement of every department is ensured by constraint 2.4. Constraints 2.5 and 2.6 ensure that the departments are within the building. Constraints 2.7 and 2.8 define the centroids of the departments.

The next set of constraints are most relevant for our study, they ensure that the departments are not overlapping each other. Constraint 2.9 ensures that department i is strictly to the west of department j (if $z_{ij}^x = 1$), if $z_{ij}^x = 0$ the constraint is satisfied if the left side of department j at the same place as the right side of department i or more to the east, so this does not prevent overlapping. Constraint 2.10 ensures (if $z_{ij}^y = 1$) that department i is strictly to the south of department j. Constraint 2.11 ensures that no two departments overlap by ensuring that department i should at least be north/east/west/south of department j. Constraints 2.12 and 2.13 ensure non-negativity and 2.14 ensures that the z-parameters are binary.

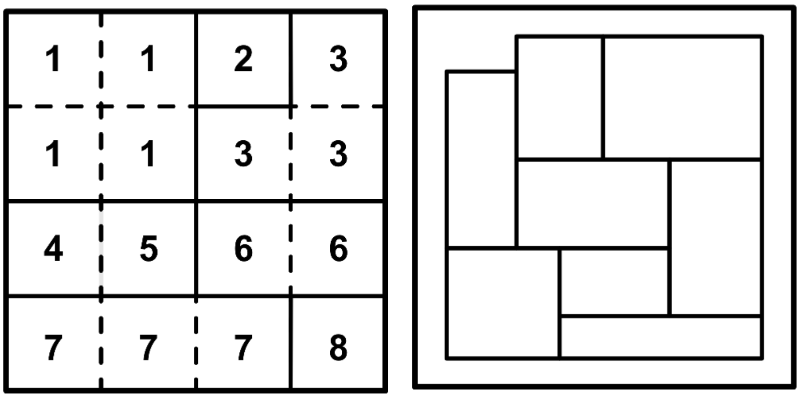


Figure 2: Discrete layout representation (left); Continuous layout representation (right)

Construction approaches build progressively the layout of the facilities until the complete layout is obtained whereas improvement methods start from one initial solution and they try to improve the solution with producing new solutions (Drira, Pierreval, & Hajri-Gabouj, 2007).

Arya, Garg, Khandakar and Meyerson (2004) state that the facility layout problem is a hard combinatorial optimization method, meaning that it can be time consuming to find improved solutions. A quick way to find better solutions in the neighborhood of the created layout is local search. Arya et al. (2004) state is that the exchange-heuristic is a popular local search. In the exchange-heuristic you swap attributes, like departments or machines, around. A problem of local search is that it can lead to a local optimum. Mavridou and Pardalos (1997) state that simulated annealing can be used to escape from this local optimum. Mavridou and Pardalos (1997) further state that annealing refers to a process of cooling material slowly until it reaches a stable state. Starting from an initial state, the system is perturbed at random to a new state in the neighborhood of the original one, for which a change in the objective function value takes place. If the optimization is minimizing, the transformation to a new state is accepted if the change is negative (so it is a reduction). If the change is positive, the transformation is accepted with a certain probability;

$$p(\Delta) = \frac{-\Delta E}{k_b T}. \tag{2.15}$$

T is the control parameter corresponding to the temperature of the cooling material. During the course of the algorithm *T* is decreased, thus reducing the probability that a new state that did not yield a better solution is accepted. Kirkpatrick, Gelatt and Vecchi (1983) state that using simulated annealing one can avoid methods that lead to locally optimal solutions and eventually higher quality solutions can be obtained. Chiang and Chiang (1998) state that in their research solutions generated with Simulated Annealing only deviate 1-2% from the best-known solution.

Muther (1961) developed a model to aid the facilities planner in developing alternative layouts. He named this Systematic Layout Planning (SLP). At its foundation stands the activity relationship chart (Tompkins, White, Bozer, & Tanchoco, 2010). A sample of the chart is given in Figure 3. The chart shows the relationship between two departments in both their importance (with a letter) and the reason (with a number). This chart gives insight in which departments have an important relationship and should be placed close to each other.

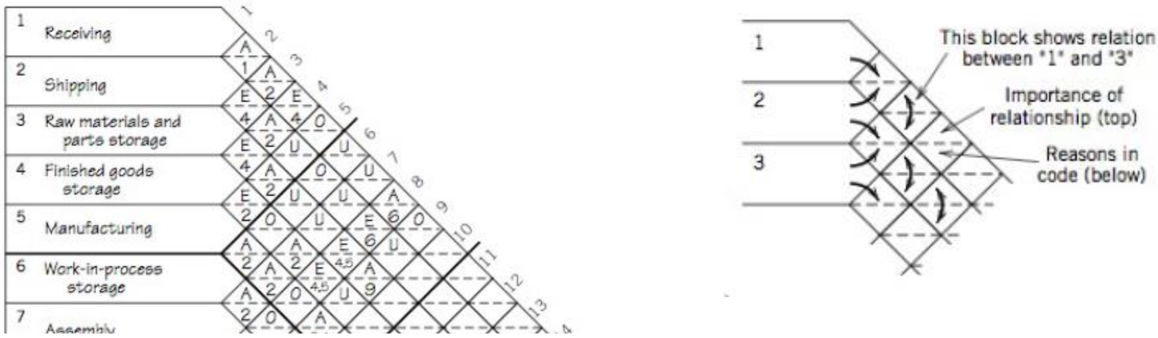


Figure 3: Example of an Activity Relationship Chart (left) and the legenda (right) (Tompkins, White, Bozer, & Tanchoco, 2010)

2.2 Material flow

This section will discuss literature about how the material flow can be expressed.

Drira et al. (2007) state that an important consideration in the design of a manufacturing facility is to determine the flow of materials, parts and work-in-process inventory through the system. The flow shows how the product, while it is being transformed from raw material to (semi-)finished product, goes through the facility from beginning to end. According to Thompkins et al (2010), the flow in a facility is typically a combination of the four standard flow patterns given in Figure 4, the receiving (entrance) and shipping department is usually fixed.

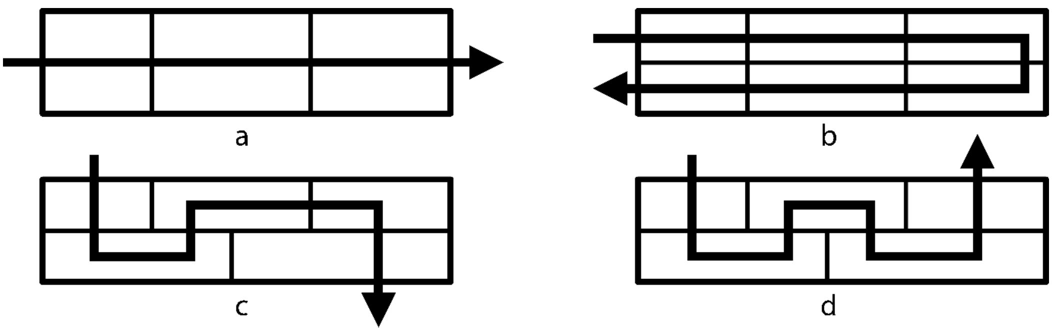


Figure 4: Standard flow patterns; a) Straight line flow, b) U-shape flow, c) S-shape flow, d) W-shape flow

Chhajed et al (1992) state that the general objective of material flow network design is to minimize the fixed cost of network construction (cost of path construction) and variable cost of flows. They have developed a material flow design network model called *Shortest rectilinear flow network problem*. This model uses rectilinear distances to calculate the distance between different stations.

Schmidt (2008) states that for identifying inefficiencies and potential savings a Sankey Diagram can be used. He further states that Sankey diagrams can be used to map value flows in systems at the operation level.

2.3 Layout Evaluation

This section will models how layouts can be evaluated.

Meller and Gau (1996) state that a layout's efficiency can be measured using material transportation and handling costs. The mathematical objective can be seen in equation 2.15.

$$c = min \sum_i \sum_j (f_{ij} * c_{ij}) * d_{ij} \qquad (2.16)$$

This equation has the following parameters: cost c_{ij} (cost of moving a unit load of material from department i to j), interdepartmental flow f_{ij} (the flow of material from department i to j) and distance d_{ij} (distance from department i to j). The material handling costs change linear with the distance the material has to move. The flow f_{ij} is a constant parameter showing how much material has to be moved in a given timeframe. To minimize the total costs, one should minimize the total distance. The distance can be measured in a variety of ways. Meller et al. (1996) give the following two ways:

- Distance between input and output points: This distance is measured between the specified I/O points of two departments and in some cases is measured along the aisles when traveling between two departments. The major drawback of this accurate measure is that one does not know the location of the I/O points until one has developed the detailed layout.
- Centroid-to-centroid (CTC): When the I/O points of the departments are not known, the department centroid is used to represent the department I/O point. The shortcomings of CTC distances includes: the optimal layout is one with concentric rectangles; an algorithm based on CTC attempts to align the department centroids as close as possible, which may make departments very long and narrow. Furthermore, Francis et al (1974) state that a department that is L-shaped may have a centroid that falls outside the department.

Tompkins et al. (2010) state that the two most used metrics to measure distances between two points are rectilinear and Euclidean distances. Figure 5 shows a graphical representation of both distance metrics.

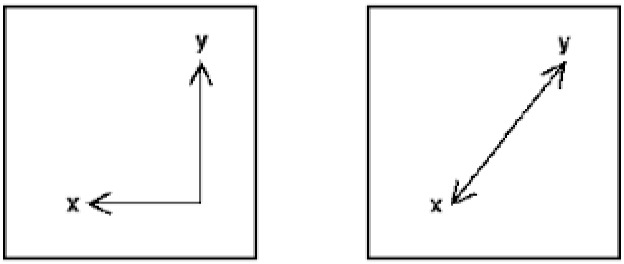


Figure 5; Left) Rectilinear distance. Right) Euclidean distance

The rectilinear distance metric measures the distance between two points along a grid. This grid has strictly horizontal and vertical lines with 90° angles between them. Rectilinear is mostly used when travel is done along paths parallel to a set of orthogonal axes. The Rectilinear distance formula is as follows:

$$d = |x_2 - x_1| + |y_2 - y_1| \qquad (2.17)$$

The Euclidean distance metric measures the distance between two points in a straight line. It is mostly used when there are no obstacles, like in air travel. The Euclidean distance formula is as follows:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (2.18)$$

Meller et al. (1996) also state that in multi-floor facility layout problems, one should also consider the vertical distance in addition to the horizontal distance. Multi-floor problems require the user to specify potential lift locations and the cost to move one-unit load one vertical distance unit between departments i and j (c_{ij}^{V}) as well as the horizontal material handling costs (c_{ij}^{H}). The mathematical objective can be seen in equation 2.18.

$$c = min \sum_{i} \sum_{j} (c_{ij}^{H} * d_{ij}^{H} + c_{ij}^{V} * d_{ij}^{V}) * f_{ij} \qquad (2.19)$$

This equation has the following parameters; cost c_{ij}^{H} (cost of moving a unit load of material horizontally from department i to j) and c_{ij}^{V} (cost of moving a unit load of material vertically from department i to j), distance d_{ij}^{H} (horizontal distance from department i to j) and d_{ij}^{V} (vertical distance from department i to j) and the interdepartmental flow f_{ij}.

2.4 Simulation

In this section, we will discuss simulation as a technology used in research about production facilities. We will define simulation in the context of our study and explain different types of simulation.

There are various techniques to understand a production system and its performance, simulation is one of them. Law (2015) defines a system as a collection of entities. He further states the following: "Simulation modelling is an excellent tool for analyzing and optimizing dynamic processes. Specifically, when mathematical optimization of complex systems becomes infeasible, and when conducting experiments within real systems is too expensive, time consuming or dangerous."

There are different definitions for simulation, but the most used is that of Shannon (1975). He states the following: "*The process of designing a model of a system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of the system*."

Model design

The first part of Shannon's definition talks about the design of the model. According to Law (2015) a system can be described in a mathematical model. This will represent the system in terms of logical and quantitative relationships that are then manipulated an changed to see how the model reacts. Above we already mentioned that systems have states, Mes (2017) defines a state as "*A collection of variables necessary to describe a system at a particular time*". Law (2015) gives three opposites to distinguish different types of simulation, namely:

- *Static versus dynamic*: A static simulation model represents a system at a particular time, whereas a dynamic simulation model represents a system as it evolves over time.
- *Deterministic versus stochastic*: Deterministic simulation models do not contain any probabilistic components, in other words do not have randomness. Stochastic simulation models can have random variables, since stochastic models produce output that is itself random, it must be treated as an estimate of the true characteristic.

- *Discrete versus continuous*: Systems can be discrete or continuous. For discrete systems, the state variables change instantaneously, where for continuous systems the state variables change continuously with respect to time.

The simulation type that fits our research is a dynamic stochastic discrete event simulation. Our simulation can then cope with the stochastic nature of input data.

Mes (2017) state that in simulation time can be handled in two different ways. Firstly there is Time-Oriented simulation, within this simulation time evolves continuously. This represents the real world the most, the time a part takes to cover the system is continuous. Another way time can pass in simulation is with discrete event simulation (DES). Within this a simulation jumps from event to event. An event can be a part entering a station, the process starting or ending.

A problem with the models given in the existing literature is that it can only cope with one input or output point per (usually) department. An advantage of simulation is that more and precisely placed input or output points can be used and the distance between them.

Validation and verification

Martis (2006) states that no model can be accepted unless it has passed the tests of validation. He further states that the validation process usually is model based and dynamic, but that a researcher can follow a methodical procedure to authenticate his model. Sargent (1994) states that validation is substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model. He further states that there are three basic decision-making approaches to determine a simulation models validity. The first approach, and most commonly used, is that the researcher or the development team makes the decision if a model is valid themselves. The second approach is called Independent Verification and Validation (IV&V). This approach uses an independent third party to validate the model. The last decision-making approach is to use a scoring model to determine a model's validity. Scores are determined subjectively when conducting various aspects of the validation process.

Sargent (1994) states that the best model verification and validation process relates to the model development process. Figure 6 shows Sargent's graphical representation of the model development process and its relationship to validation and verification. It starts with the *Problem Entity*, this is the idea, situation, policy or phenomena to be modelled. The *conceptual model* is the mathematical, logical or verbal representation of the problem entity. The *computerized model* is the conceptual model implemented on a computer. The conceptual model is developed through an analysis and modelling phase, the computerized model is developed through a computer programming and implementation phase.

Sargent (1994) then explains the relationship between the validation/verification and the modelling process. *Conceptual model validity* is defined as determining that the theories and assumptions underlying the conceptual model are correct and that the model represents the problem entity. *Computerized model* verification is defined as ensuring that the computer programming and implementation of the conceptual model is correct. *Operational validity* is defined as determining that the model's output behavior has sufficient accuracy for its intended purpose. *Data validity* is defined as ensuring that the data necessary for model building, model evaluation and testing are adequate and correct.

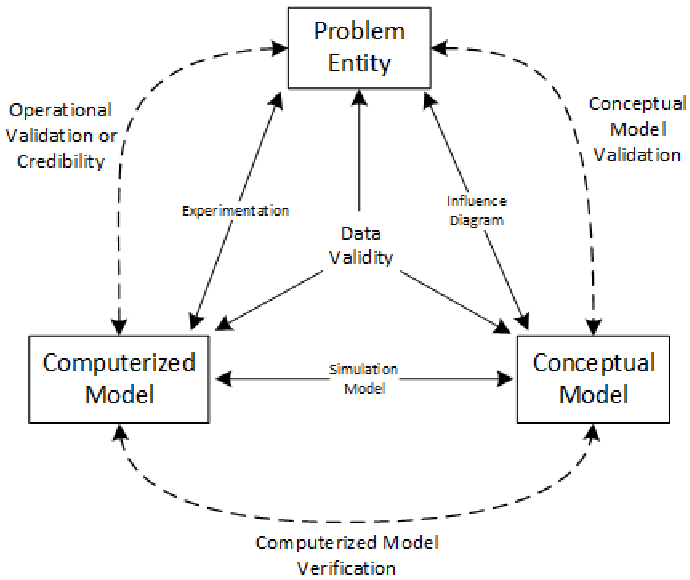


Figure 6: Simplified version of the Modelling Process (Sargent (1994))

Conceptual Model Validity is determining that the theories and assumptions underlying the conceptual model are correct, that the model representation of the problem entity and the model's structure are "reasonable" for the intended purpose of the model. *Computerized model verification* is ensuring that the computer programming and implementation of the conceptual model is correct. *Operational validity* is primarily concerned with determining that the model's output behavior has the accuracy required for the model's intended purpose. A list of applicable validation techniques is given in the next section.

Sargent (1994) presents eleven different verification and validation techniques. The techniques can either be subjective or objective, usually a combination of these are used. The list of techniques can be found in the Appendix.

2.5 Conclusion of literature study

This section will conclude the literature study.

At the end of our literature study we answer research question two. We have discussed the current models for facility layout problems and material flow and found what is missing in them, namely that one can only have one input or output point and that distances are approximated using either Euclidean or Rectilinear distances. We are going to create a dynamic stochastic discrete event simulation model for our research. With this model we can quickly and cost efficiently investigate different layouts. A big advantage of a simulation model is that we can use the true distances from multiple output to input points, instead of the less precise Euclidean or Manhattan distances. We will furthermore use the non-overlapping constraints of the model introduced by Tompkins et al. (2010). The simulation model validation and verification steps mentioned by Sargent (1994) will be used to validate the model.

3 Analysis of the existing locations

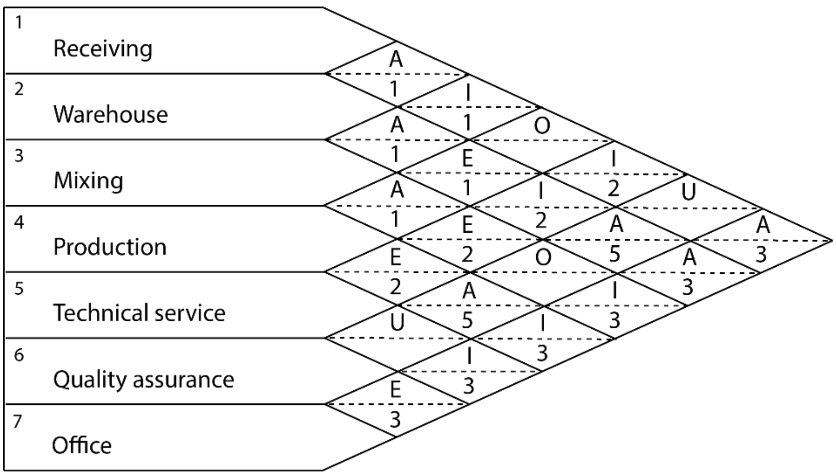
To have a better understanding of the current situation at all three existing production facilities of PaperFoam, we have analyzed the work that is done at every facility and more specifically at every department within those facilities. Section 3.1 will give a general introduction to the activities done in all three locations. Section 3.2 will explain the methods used to collect the required input data from the production locations. Section 3.3 explains the specific input data for the Dutch facility in Barneveld. Section 3.4 explains the US facility in Leland and section 3.5 explains the Malaysian facility in Penang.

3.1 Overall layout

This section will explain the overall layout of all three locations of PaperFoam.

At all three PaperFoam production facilities there are 6 main departments. Namely the mixing area, the production area, the technical service area, the quality assurance area, the warehouse and the office area. The activities per department will briefly be introduced. We will then clarify per location for all non-deterministic activities how we measured the times and what statistical distribution corresponds with each activity. All deterministic processes, like the machine settings are set and given by PaperFoam.

Figure 7 shows the activity relationship chart with next to it the legends. We created this chart based on Muther's Systematic Layout Planning (SLP). This chart shows that there is a lot of material being moved from the receiving area to the warehouse. All three of PaperFoam's current locations have the receiving area within their warehouse. There is also material being moved from the warehouse to the mixing area, which has its own small area for temporary storing raw materials. Another important flow is from the mixing area to the production area, this will be the main focus of this study. The office has mainly the purpose of informing all other departments, hence the information value.



Value	Closeness
A	Absolutely necessary
E	Especially important
I	Important
O	Ordinary closeness
U	Unimportant

Value	Closeness
1	Flow of material
2	Service
3	Communication
4	Same personnel
5	Flow of product

Figure 7: Activity relationship chart (left) and the legenda's (right)

Mixing area

The production process starts in the mixing area. This is the place where the raw materials are mixed into a batter. Every batch consists of approximately 150 kg dry material. The dry material consists of industrial starch (mostly from potatoes), paper fiber and the company's secret premix. This is then mixed with water and natural coloring. This batter is then pumped into transport vessels. The people working in the mixing area will then bring the full vessels to several locations in the production area.

Mixing consists of the following steps:

1. Cleaning the mixer.
2. Filling the mixer with the required amount of water.
3. Adding paper fiber and coloring into the mixer
4. First mixing stage for 20 minutes (30 in Malaysia)
5. Adding starch and premix
6. Second mixing stage for 20 minutes (30 in Malaysia)
7. Filling the transport vessels (approx. 50 kg dry material per vessel)

Per location we measured several things. The time required to clean the mixer, the time required to fill the mixers with water (and how much water was needed), the time required per step and the total time required per batch. The amount of batter per transport vessel differs a bit, since the vessels are manually filled. The results of these measurements are given per location in sections 3.3, 3.4 and 3.5 respectively.

Production area

The production area consists of different production lines and every line contains 6 machines. Every machine is an injection molding machine with one mold. The number of products the machine produces per stroke is depending on the amount of cavities in the mold. The number of cavities per mold depends on the size of the product. Every machine has its own vessel connected or gets the batter from a shared prefoamer. The prefoamer adds extra air into the batter to improve the molding process. The batter is then moved to the connected machines using pipes. The amount of batter needed per machine stroke depends on the size of the product (and the number of cavities). The amount of material needed is deterministic and controlled by the machine.

After every machine stroke, the manufactured products are dropped on a conveyor belt that moves them to an operator that removes the overshoot and packs the products into boxes. The overshoot is the extra material that is left around the finished product from the injection molding process. The amount of time that these steps take are measured and given per location, in Sections 3.3 to 3.5. We have also measured what percentage of products are rejected; this is important to accurately predict the output frequency of full boxes.

Technical service

Besides making sure that all the machines keep working, the technicians are responsible for two more things, namely changing the molds and swapping the empty vessels with full vessels to the machines. They keep walking through the facility to check how much batter there is still in the vessels and change it if it runs below approximately 10%. PaperFoam has performed a study to get an insight in how much variation there is in material left after they are taken off the machine. This is on average 8.6 liters with a standard deviation of 2.1 liters.

Quality assurance

The quality assurance department has the responsibility to check the outgoing finished products. They randomly select an amount of product per box that has to be checked on faults. The number of products they have to check is given by military grade quality assurance standard index. After they sign the boxes off, they will be moved to the warehouse.

Receiving

Incoming raw materials will be checked in the receiving area. At all three current locations of PaperFoam, the receiving area is part of the warehouse.

Warehouse

The warehouse stores incoming raw materials and finished products. For this study, the movement within this department is ignored, so we also did not do any measurements here.

Office area

The office area is mainly for the support staff of the facility. This department is not interesting for our study, but we should take into account that some space should be reserved for this area.

3.2 Measurements and method

During our data collection phase, we have measured different steps within the production process of PaperFoam. We will briefly mention what we measured and how. After that we will present the findings per locations.

Mixing area

The mixing process was relatively straightforward to measure. The steps the mixers have to take are the same for all different recipes that they have to make. Depending on the location, some mixing times where different, so we spend some days per location within the mixing area and noted down all times at which certain steps started or ended. Overall, we have noticed that the total time that mixing a batch of batter took highly depends on the human factor. Most of the extra time, where the mixing machines stood still, the operator responsible to make the batter were elsewhere occupied or did not pay attention that the machine was finished with its stage. Furthermore, we have measured how fast technicians or the employees making the mixes can walk with transport vessels. We measured this by defining a stretch of 5 meters, where the mixer could walk freely, and recorded multiple times how long it takes to cross that distance.

Production area

The task of the operators on the production lines was more challenging to measure. The operators perform a series of tasks that they do in batches and either one task per person (in Malaysia) or several tasks at the same time. For this reason, we took recordings of their work so that we can analyze frame for frame what task they were doing, how many products they did that to at the same time and how long it took.

Technical service

The workers from the technical service have two main tasks. The changing of vessels on the machine or molds in the machine. Both tasks have the effect that the machine is temporarily not producing, were also measured by noting down the time a task was started and ended.

3.3 Barneveld, The Netherlands

This section will contain information specific for the Dutch facility.

The production facility in Barneveld, The Netherlands is the oldest of the three locations. Production started here in 2011. Figure 8 shows the floorplan of the facility in Barneveld. It consists of two production areas, divided by a wall, and the mixing area is next to the second (the top area in the floorplan) production area. This second area is a newer addition to the building. The arrows show the flow from the transportation vessels from the mixing area to the production lines.

Floorplan

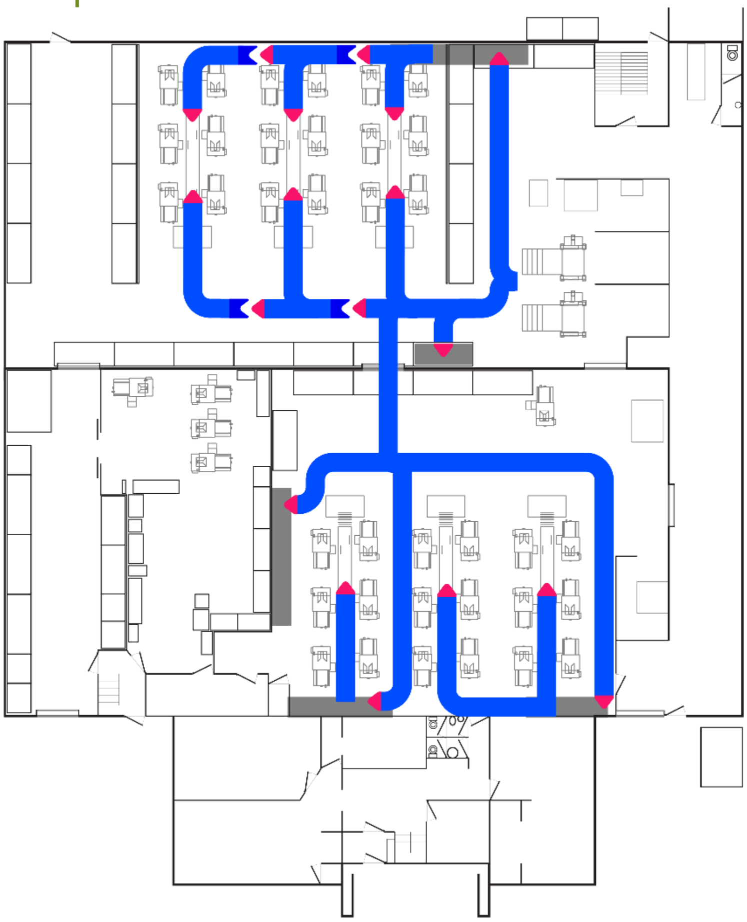


Figure 8: Floorplan of the Barneveld location (with flow of vessels from mixing area)

Mixing area

We have measured 39 different batches. The shortest total measured time was 56 minutes and the longest was 2 hours and 36 minutes. Figure 9 shows a histogram with the distribution of measurements divided in bins of 10 minutes. The high outlier can be explained by the pump braking, so this can then be seen as an exception. The mixing times varied so much and had so many external factors that we decided to use the data as historical data and sample a time random to determine the mixing time. The mixer walked with a transport vessel on average 1.4 meter per second. The mixer could be filled with 150 liter per minute; this is way higher than at the other two locations because in The Netherlands, PaperFoam uses a buffer-container that is filled with water. The other two locations rely on water pressure, which drops if for instance barrels are being cleaned.

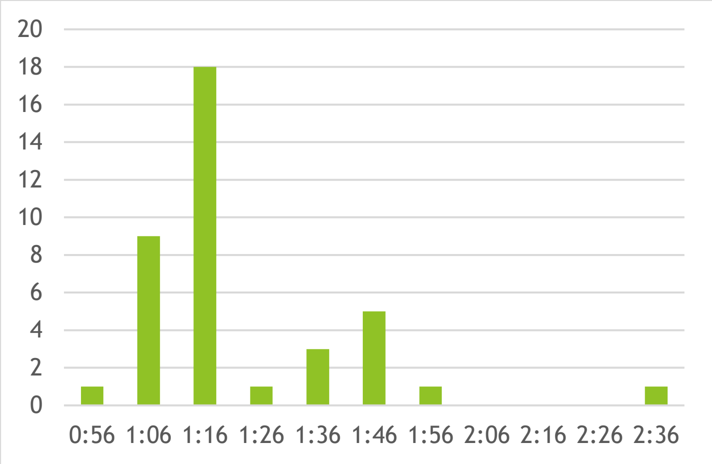


Figure 9: Histogram of measured mixing times

Production area

We have analyzed 14 hours of recordings and found that on average the operators need 6 seconds per product with a standard deviation of 2 seconds.

Technical service

We measured 27 mold changes and found that on average it takes a technician 4 hours and 29 minutes with a standard deviation of 26 minutes. This is measured from when they start by undoing the first bold, so the machine is shut down earlier, until the technician gives the machine back to production. This means that the machine produces the right product without interference of the technician. Figure 10 shows the histogram of the recorded times, the bins are steps of 10 minutes.

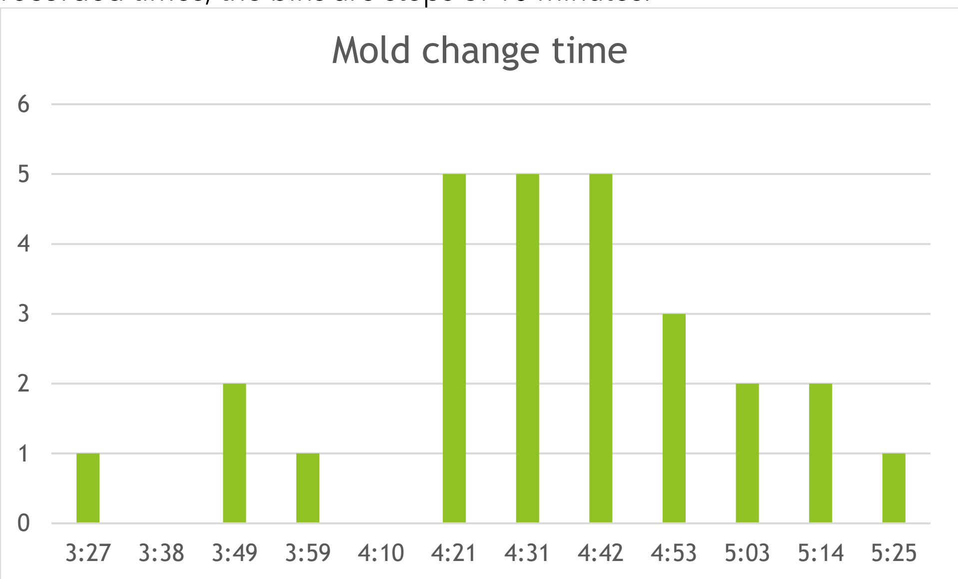


Figure 10: Histogram of measured mold changing times

3.4 Leland, USA

This section will contain information specific for the American facility.

Floorplan

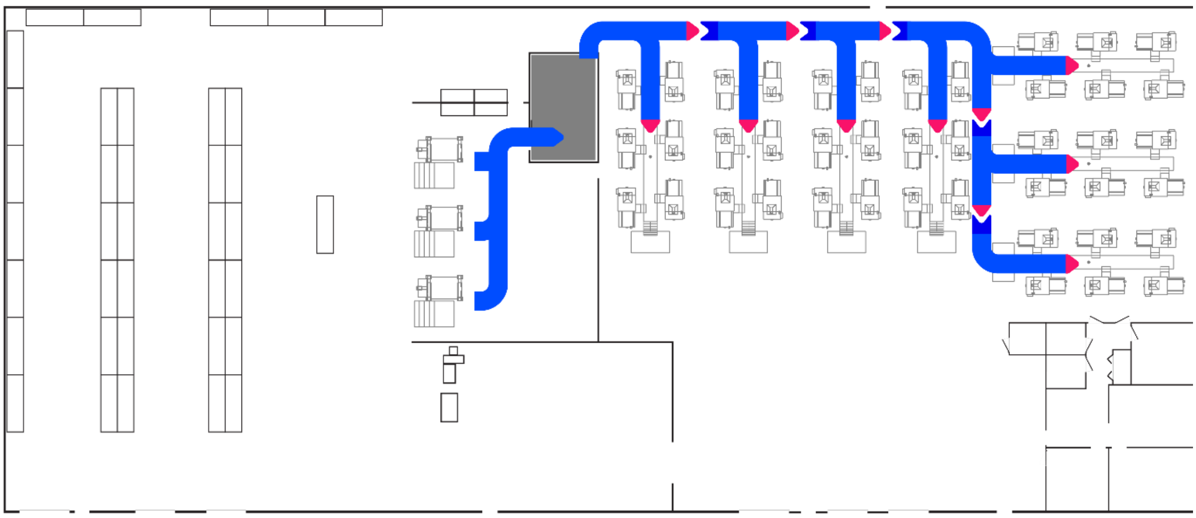


Figure 11: Floorplan of the Leland location (with flow of vessels from mixing area)

Mixing area

Because the facility was running a bit slower than usual when we were there, we were not able to do as many measurements as wanted. In total, we measured the times of 14 batches. The shortest time measured was 1 hour and 33 minutes and the longest was 2 hours and 58 minutes. The mixer is filled with a speed of 15.2 liter per minute. Figure 12 shows the histogram of the measured mixing times divided in 10-minute bins.

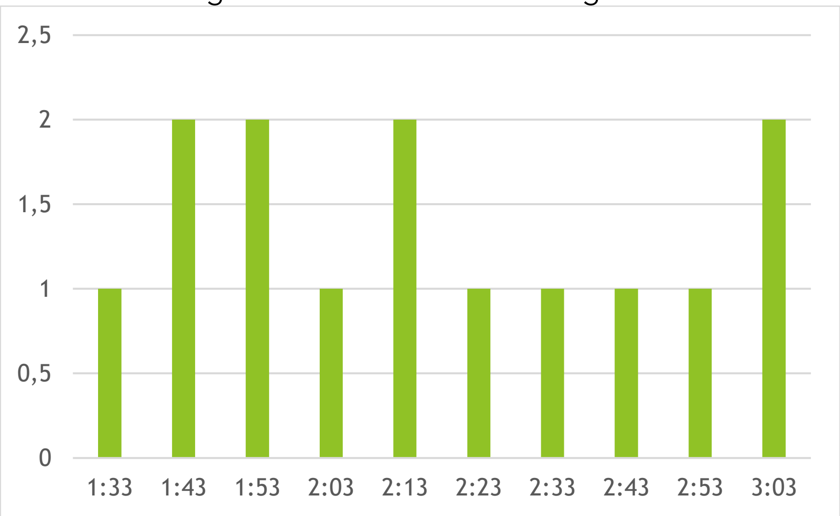


Figure 12: Measured mixing times in Leland

Production area

Due to regulations, we have not made recordings in the US. We did do 20 manual observations and compared these to the results gathered in the Dutch facility. We found no statistically significant difference between both datasets. Both locations also have the same quality standard, so we can use the same distribution as used for the Dutch facility.

3.5 Penang, Malaysia

The next section will contain information specific for the Malaysian facility.

Floorplans

The Malaysian facility consists of two separate buildings, both buildings have two floors. The top floor is used for production. In the ground floor of the first building the mixing machines are placed. The ground floor of the second building is the warehouse.

Building 1:

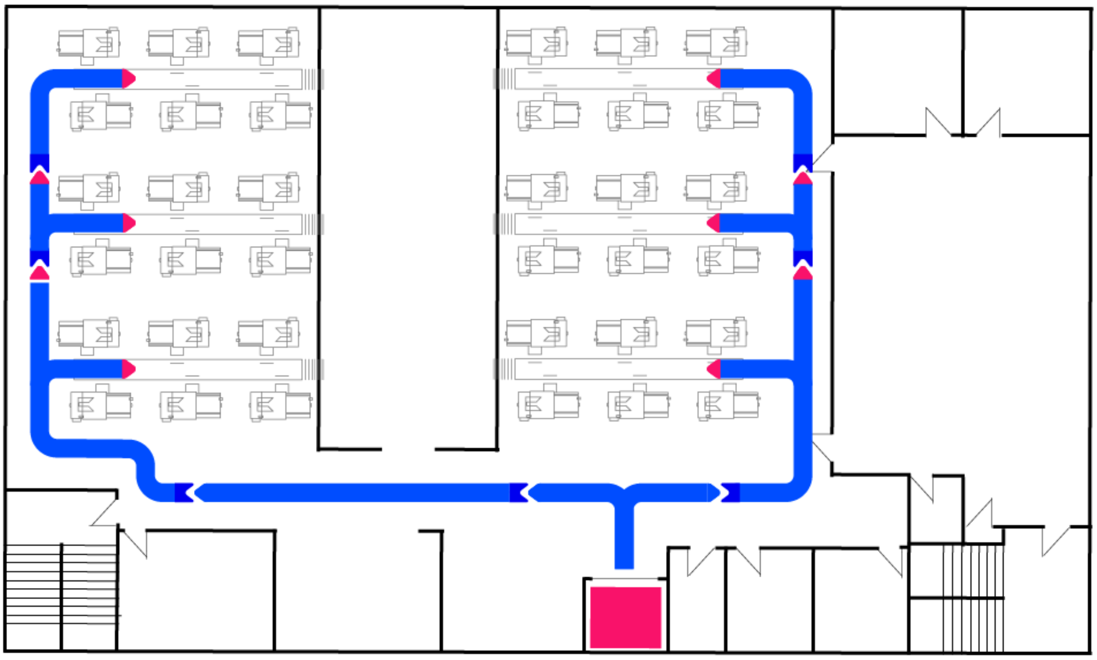


Figure 13: Floorplan of the upper floor of building 1 of the Penang location (with flow of vessels from mixing area)

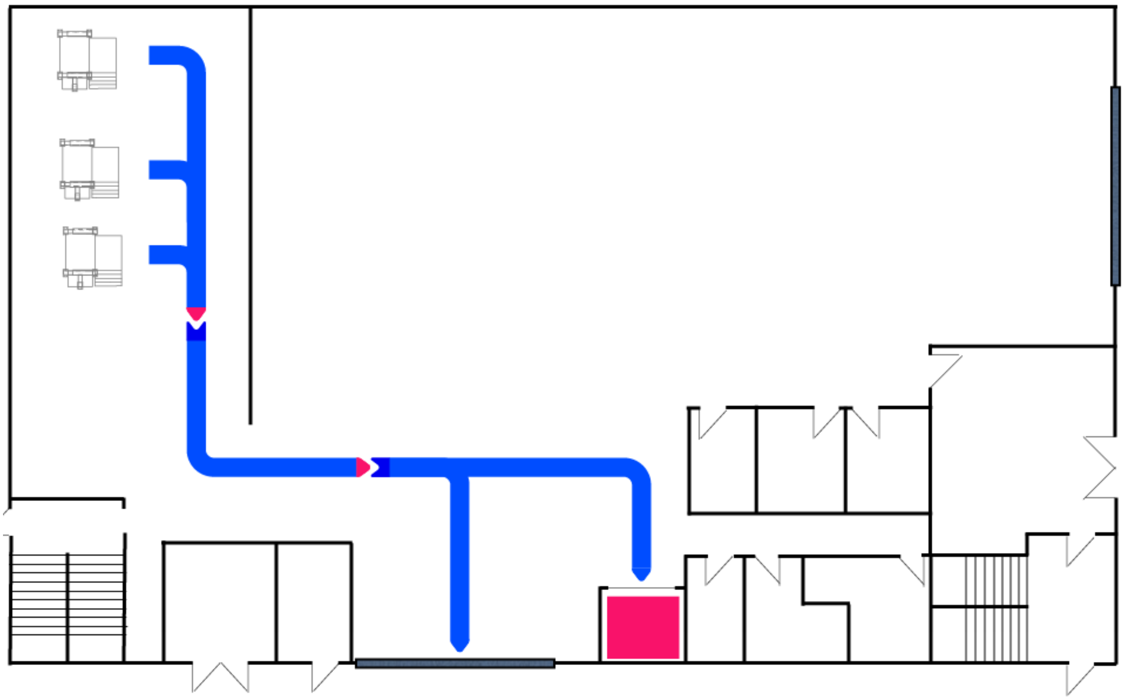


Figure 14: Floorplan of the lower floor of building 1 of the Penang location (with flow of vessels from mixing area)

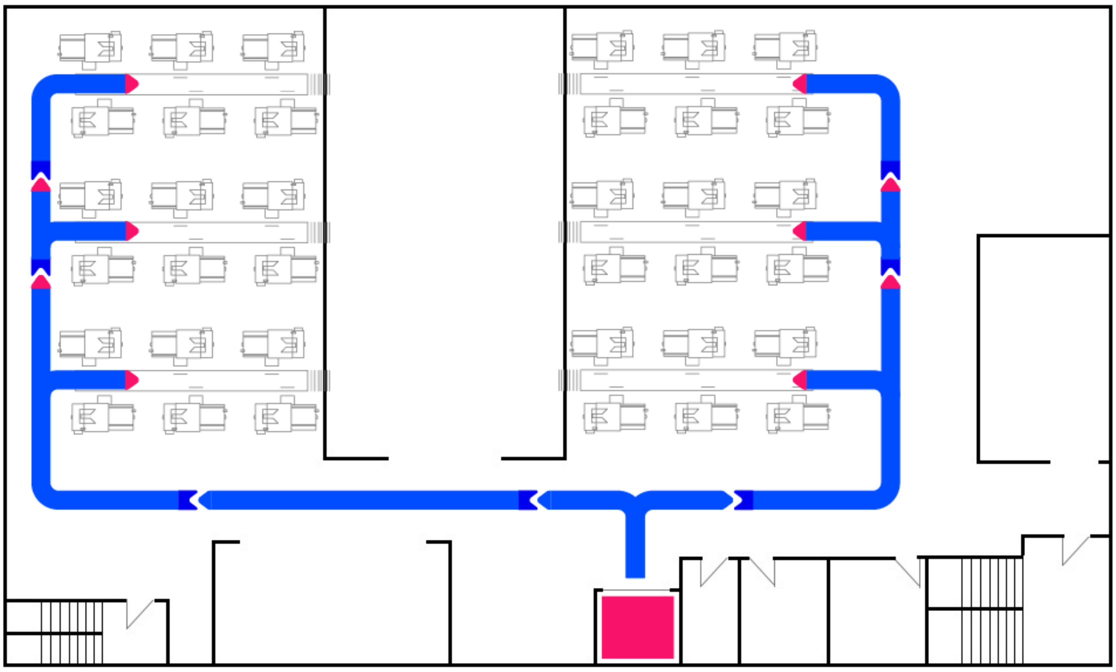


Figure 15: Floorplan of the upper floor of building 2 of the Penang location (with flow of vessels from mixing area)

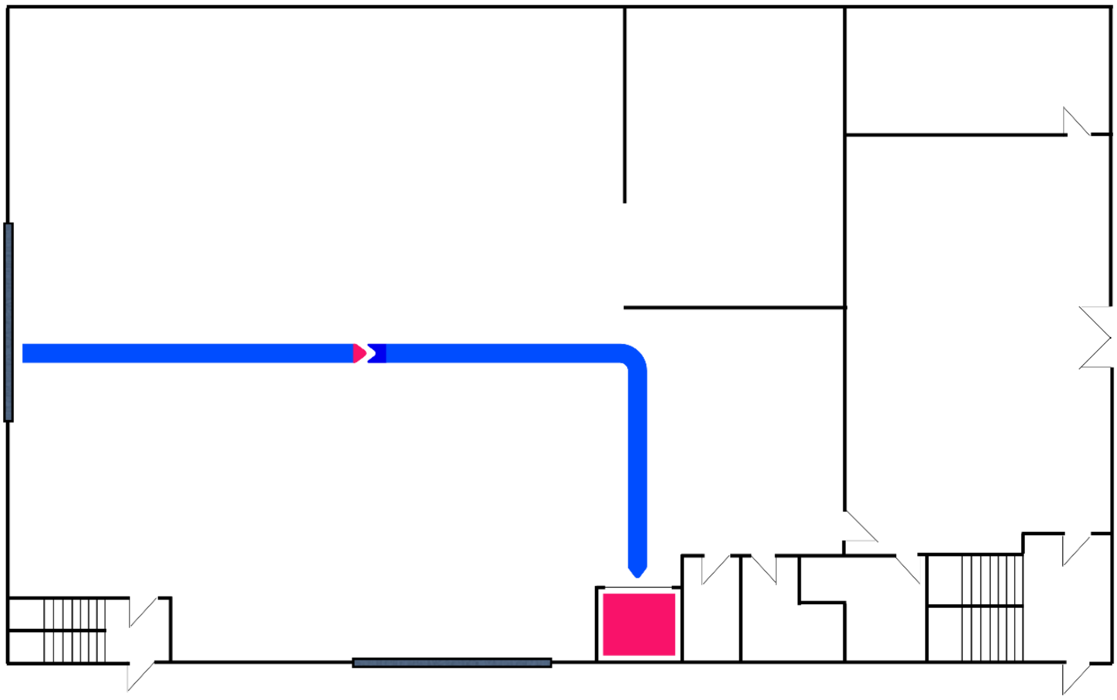


Figure 16: Floorplan of the lower floor of building 2 of the Penang location (with flow of vessels from mixing area)

Mixing area
In Malaysia, we measured 29 batches. The shortest batch was 1 hour and 47 minutes and the longest was 3 hours and 47 minutes. Since the mixing times vary a lot and are heavily influenced by different factors we have decided to use all measured times as historical data and sample a mixing time. The mixer is filled with a speed of 17.8 liter per minute. On the first floor, the vessels are moved with a forklift and on the second they push the vessels. The forklift moves with an average speed of 2.8 meters per second and the vessels are being pushed with an average speed of 1.2 meters per second. Figure 17 shows the histogram of the measured mixing times divided into 10-minute bins.

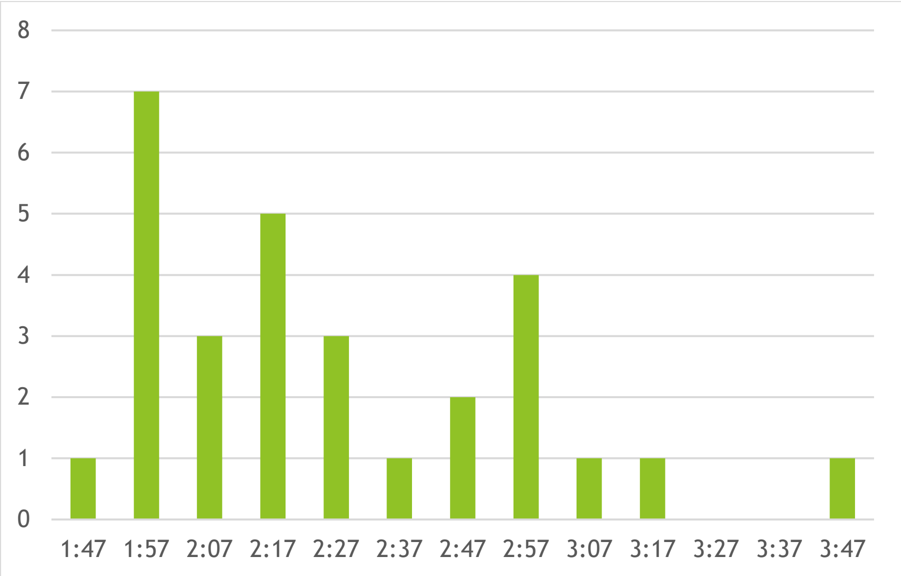


Figure 17: Mixing times of the Malaysian facility

Production area
In total we have analyzed 23 hours of material and found no clear difference in time needed per product type. Since they use on average an operator per running machine, they usually have multiple operators per production line. This means that the work is clearly separated per person and they handle every product with the same degree of precision. On average they need 5 seconds per product with a standard deviation of 3 seconds.

3.6 Conclusion
This section will conclude the analysis of the current locations.

We have analyzed the current situation in all three production facilities of PaperFoam. We found that the mixing times for the maxing machines vary a lot between the sides, but also vary a lot among each other. This is due to multiple external factors. For our simulation we will use random sampling to determine this time. Furthermore we found that the time an operator needs to clean the finished product of overshoot left by the production method (injection molding) is relatively stable and not determined by location or product.

4 Layout optimization model

This chapter will explain how and why we created a general model to generate optimal layouts for production facilities using simulation. Section 4.1 will explain the goal that we try to achieve. Section 4.2 will introduce the method we will use to create the new layouts and how to evaluate them. It will also explain what we do and do not include into the simulation and why. Section 4.3 explains what input information is needed. Section 4.4 will explain the output the model will give.

4.1 Goal

This section will introduce the goal of our research.

As mentioned in section 1.3, depending on the location, 20 to 50% of manufacturing costs are due to handling of parts. A big part of this is the interdepartmental transportation. To easily and quickly get inside into the costs of current layouts and to create new layouts, a general model will be created. This model can be used by production facilities that want to minimize interdepartmental flow by optimizing their layout. The model will focus on reducing the necessary movement of the interdepartmental flow. Since the main goal of our research is to minimize the time an operator has to walk with a vessel, we have to minimize the distance they have to travel.

4.2 Method

This section will introduce the methods that are used to generate new layouts. It will also introduce the scope of the research and the assumptions that have been made. The approach on how to determine certain key aspects, like the path an operator will travel, will be clarified.

Testing different layouts in real life would be very expensive, so the model will entail simulation to evaluate the generated layouts. We have chosen to use computer simulations, since we can more realistically calculate distances between two objects. The standard methods use either rectilinear distance or Euclidean distance. Both do not take obstacles, like walls or other attributes, that an operator has to walk around into account. The model should work for both improving existing layouts as for creating layouts for new facilities. For existing facilities, we will use a coordinate system to define where all attributes are. The attributes are; the walls, the production lines, the mixing machines, quality assurance and the canteen.

If we start with an existing location, we will first simulate the current layout and determine the required costs for moving all material. New layouts will then be created using different optimization methods. These methods are designed for brownfield optimization, since the methods can handle the strict restrictions. The methods investigated are:

- For every attribute to move, generating randomly generated coordinates, between the constraints of a given area, and place them there. If one or more attributes cannot be placed, due to no room, redo the whole process, since this did not yield a feasible solution. Stop if everything is placed.
- Place the attributes along a wall. Loop through the four outer walls (north, east, south and west), and place the attributes alongside a different wall every experiment. The attributes will be placed perpendicular to the wall and sequential besides each other. Also change the side you start positioning the attributes from, for example for the north wall, start at the west side and place them to the east and start at the east side and place them to the west.

Within both above mentioned optimization methods local search will be used to find it there would have been a better solution within the generated layout. With local search we swap the production lines among each other. Since the production schedule is determined per machine and per production line, the production schedule could have an effect on the result. The production line with the longest total walk time will be swapped with the one with the shortest to remove this effect of the production schedule. This is swap is done until a search-run is rejected. To determine if a run will be rejected simulated annealing is used, this prevents stopping in a local optimum.

From the introduced two methods we formed 9 experiments to generate new layouts, with each experiment having the local search algorithm to further find improved solutions within the created layout.

Next simulation run selection
Figure 18 shows a flowchart of how the next simulation run is decided after the previous run ended. A simulation run is where we run the simulation for a predefined length, starting with an empty model. Firstly it checks whether another replication should be run, if so, reset the simulation and run the model with the same settings. If no more replications are needed, we check if the previous run was a new experiment, which means that we just created a new layout, if this is the case, we will start a local search to further improve the generated layout. If not, we check if the previous run was already a local search algorithm, this has to be the case, because a run is either a new experiment or local search, so show an error if this is not the case. If the previous run was an experiment, we check if this local search yielded a better solution, if this is true, then we save this solution. Using simulated annealing the simulation determines if the solution is accepted and another local search-run is started or if it is rejected and this experiment is finished.

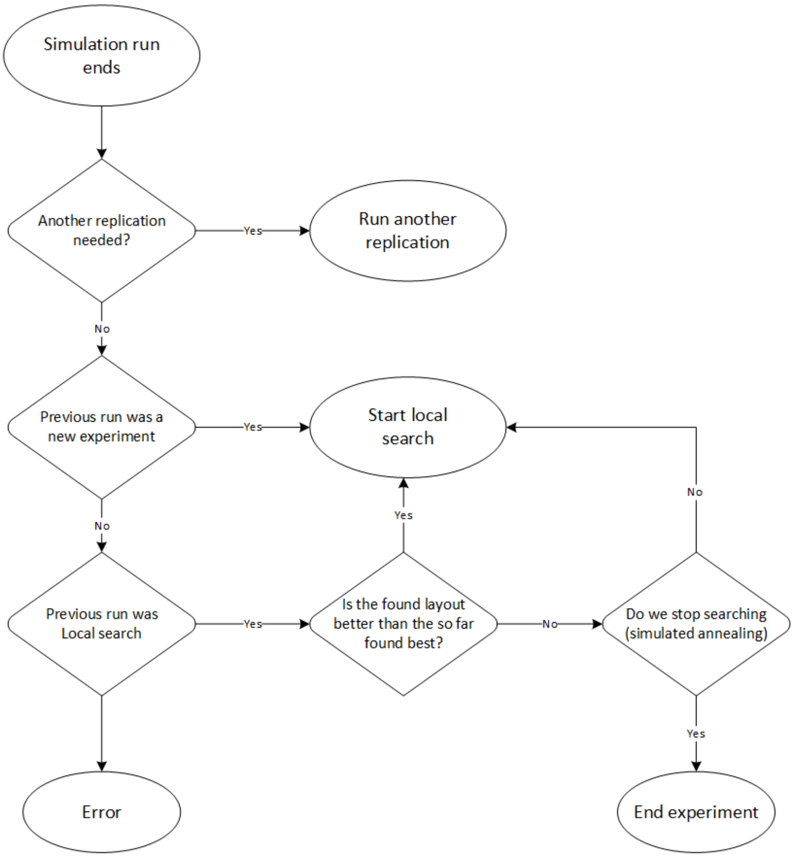


Figure 18: Flowchart of how the next simulation run is determined

Layout creation

A new layout can be created via the two developed methods. If the strategy is to create a random layout, it creates a random x- and y-coordinate between the start and end of the building. When the coordinates for the attribute are determined, the method checks if this position is valid. A position is valid if the attribute does not overlap any other attributes or walls. Appendix Figure 3 is a flowchart explaining graphically how this method works. If the position is valid, the attribute will be placed there and the method will go to the next attribute that has to be placed (if any). If the position is not valid, the method will generate a new location. To prevent this method to run infinite, it will stop after it tried to place an attribute 20 times.

If the strategy is following a wall, it checks which wall to follow, which is given in the experiment-table. Figure 19 shows a flowchart on how the "follow a wall"-strategy determines the coordinates for an attribute. After the to follow wall is determined, it then will check where to start, either left or right (for a vertical wall this means north or south). The previous used coordinates are used, if this is the first time the method is called, the start of the wall to that side will be used. When starting left, the x-coordinate (y-coordinate if a vertical wall is followed) have to increase compared to the last placed attribute, with a given step size plus half the attributes width (since we calculate the center). For example if started right, the x-coordinated is decreased with that step size. The method then checks if the attribute would be placed outside the building, which means that the newly calculated coordinates are smaller than the west wall or bigger than the east wall (or smaller than the north and bigger than the west wall). If that is the case, the y-coordinate (x-coordinate if a vertical wall is followed) is increased (or decreased if started at the right) with the step size and the x-coordinate (y for vertical) is set back to the starting coordinate.

This method ensures that all attributes will be placed next to each other as long as there is room, and otherwise they will be placed in a new row below the other.

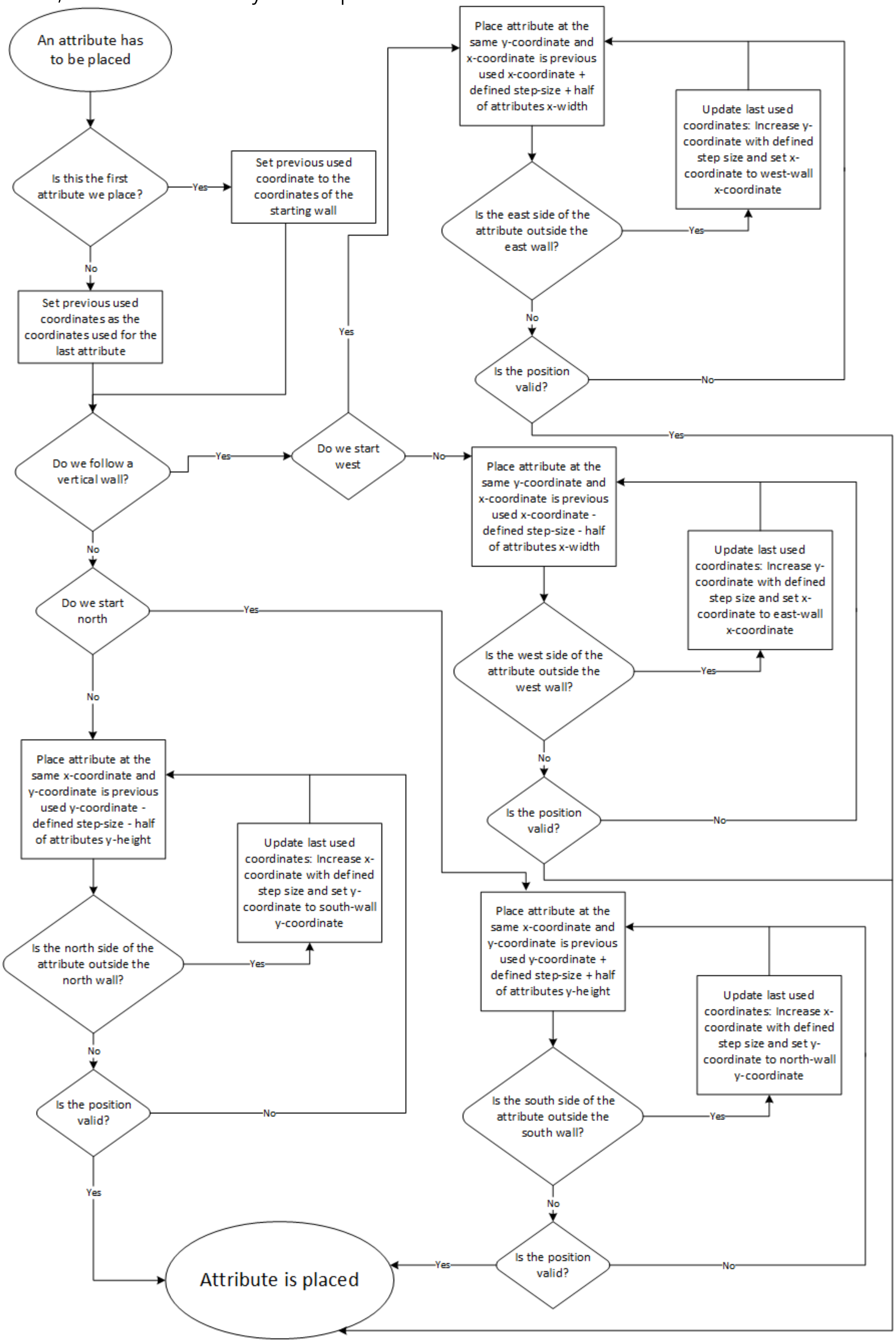


Figure 19: Flowchart showing how the coordinates for an attribute are determined

Scope and assumptions

During this study we mainly focus on the interdepartmental movement between the mixing and production area. The movement within the warehouse and quality assurance are not taken into account. We mainly focus on the interdepartmental transportation of raw materials from the mixing area to the production area, since we observed that this is the most movement within the facilities.

We assume that the parameters of the building(s) are known and given. This includes the walls, positions of the mixing machines and the position of the Quality Assurance. For our study we assume all attributes are fixed and will only change the positions of the production lines. We furthermore assume that everybody will work as predictable as possible, like no unexpected breaks or other distractions.

Approach

Prevent overlapping attributes

We will use the constraints defined by Tompkins et al. (2010) (see Chapter 2) to prevent the attributes (mainly the production lines) overlapping each other, other attributes or the walls. All center points are known, from the dimensions of the attribute we can calculate the north, east, south and west coordinates. If an attribute needs extra space for people to walk, we included it within the boundaries of that attribute. Appendix Figure 3 shows graphically how we conclude if a generated position is valid.

Obstacles

To simulate more realistic walking distances an operator has to travel from start position to destination, we need to define where the operator is allowed and not allowed to walk. We do this with so called barred areas. An operator decides its path to walk before he leaves his starting position. From this position he plots a straight path to his destination, if this path crosses an obstacle will walk around it. He will always take the shortest of the two directions. If there is no possible route for the operator to reach his destination, that layout will be flagged as invalid and another layout will be created. Constraints like the above mentioned prevention of overlapping attributes should ensure that there is always a valid path for the operator to reach his destination.

Simulated Annealing

To determine if we will do another local search-run or not we use simulated annealing. To determine if the solution will be accepted a random number between 0 and 1 will be generated. The annealing-factor will then be calculated, equation 4.1 shows the simulated annealing formula.

$$e^{\frac{-\Delta E}{T}} \tag{4.1}$$

$$\Delta E = \text{current found solution - best found solution} \tag{4.2}$$

The natural exponential function *e* is used. The power of the exponential function is ΔE divided by T. Formula 4.2 shows how ΔE is calculated, it is the difference between the solution found in the current simulation run and the best found solution within this experiment. T is the temperature-factor that is lowered with 10% every time a local search-run is accepted, thus lowering the probability that a worse solution will be accepted. If the randomly generated number is lower than the annealing-factor the solution will be accepted. If the current found solution is better than the so far found best solution, the annealing-factor will be above 1, thus will always be accepted.

4.3 Input

This section will explain what input information is needed to feed the model.

Walls of the building(s)
The walls have to be given in a coordinate-system. With the starting and ending x- and y-coordinates we can determine where to draw the wall. This will be done in 2D and 3D. The 3D will help to make the walls an obstacle for the walkers, so that the path from their start position to their destination will be more realistic.

Number of buildings
If a production facility has more than 1 building or floor the relationship between these buildings or floors have to be given. A list will provide the connection between them and the time it takes to travel between them.

Production schedule
A historically accurate production schedule will be used as input for the simulation. This schedule defines which products will be produced per production machine on a given day and shift. During the validation of the simulation model we will use tracing to determine if the simulation gives the expected output results. During the optimization we will sample what product to create per machine from the historical data. For this we select a random day and shift, but the same machine, from the schedule. Since production is dependent on the day of the week, for instance the production is quieter in the weekend, we do sample for the same day of the week. So if a new shift starts, for instance the morning shift on a Wednesday, we pick a random shift on a random Wednesday out of the given historical production data.

Worker schedule
A historically accurate personnel schedule will be used. This ensures that a realistic number of operators work at a given shift and day. Operators can either mix the batter or pack the finished product. We have divided these two tasks, calling the operators that mix the batter "mixers" and the others "operators". The number of operators working per production line is also defined.

Production lines
For the production lines it is important to know how many production lines there are and what their exact locations in the coordinate-system are. The number of machines per production line can differ, this has to be clarified. The production lines will be identified with a number, the machines in them as well. For instance machine number 1 in production line number 2 will be called machine 2.1. The work place of the operator, a table, can either be parallel or in line with the conveyer belt of the production line, this has an impact on the footprint of the production lines, so has to be defined.

Mixing machines
For the mixing machines it is important to know how many there are in the production facility and their location in the coordinate-system. PaperFoam has two different designs for their mixing machines. They differ in how the stairs leading to the drum are positioned, this results in a different footprint. The stairs are either in line with the drum or parallel to it. The location where the empty vessels have to be brought also has to be defined in the coordinate-system. Figure 20 shows a top view of the two different designs.

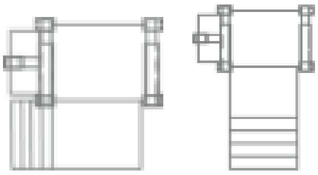


Figure 20: Top view of two different designs of mixing machine

Shift calendar
The shift calendar defines how many shifts there are, when they start and when they finish. It also clarifies when breaks start and finish. At the start of a break all operators will return to the canteen. At the end of the break they will return to the task they were performing before the break.

4.4 Output
The model should output the total amount of products produced and the total time all operators had to walk between the mixing machines and the production lines. The number of products produced can be used to verify the model. The total time all operators that move vessels have walked is the Key Performance Indicator (KPI) we will use to test if a given layout solution is better than the original layout.

5 Software implementation

The previous chapter explained the conceptual model, this chapter will explain how we translated that model into a computer model.

The simulation will be non-terminating, which means that there is no defined end point. The simulation will be built in a software tool named Siemens Plant Simulation version 13.0, which is a software tool to build and simulate models. Plant Simulation is object orientated. The program has pre-programmed objects, which can be used to build a model. These standard objects can be customized with user-defined attributes, variables and programmable methods. This allows the designer of the model to tailor the simulation to the specific problem or process. Plant Simulation uses discrete event simulation (DES), which means that it needs triggers (events) to start certain methods or process flows. In Plant Simulation everything starts with a Frame object. Within these Frame objects different attributes like Material Flow and Information Flow objects can be placed. Material Flow objects are objects like singleProcs, buffers, lines and sources/drains. SingleProcs are processing stations with 1 workstation. If there are more processes that can be worked on in parallel a ParallelProc is used and the number of parallel processes should be defined. Buffers will hold MUs (movable units) until the station it connects to has room to allow a new MU. Lines are like conveyer belts where MUs moved over with a predefined speed. Sources create MUs at a predefined interval and drains remove MUs.

5.1 The simulation model

This section shows all frames used in the simulation model and explains their function and contents. It will first introduce the MUs used, it then will give a short introduction into the frames, the more in-depth description can be found in the Appendix Chapter 2.

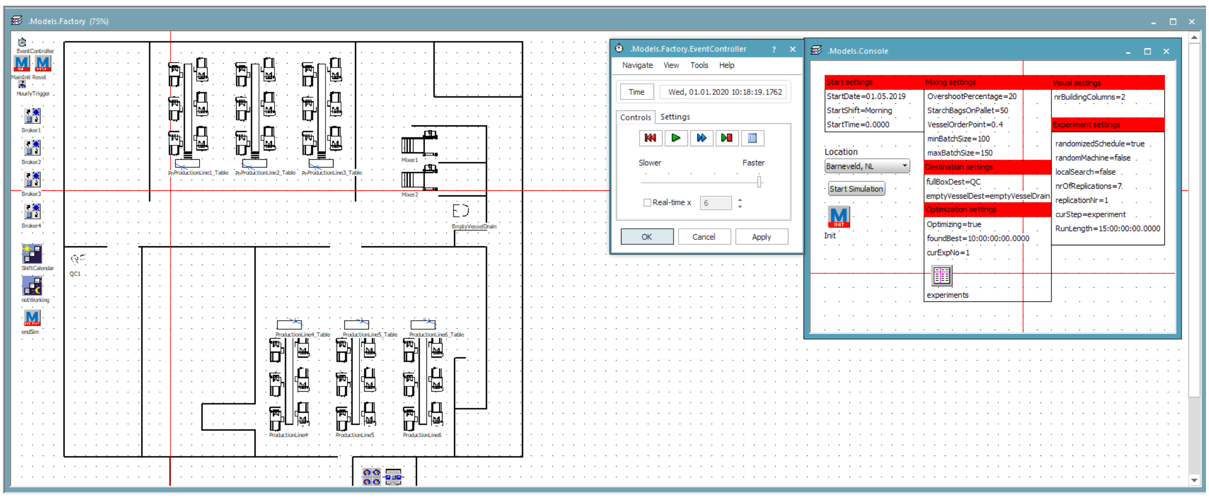


Figure 21: Screenshot of the simulation model

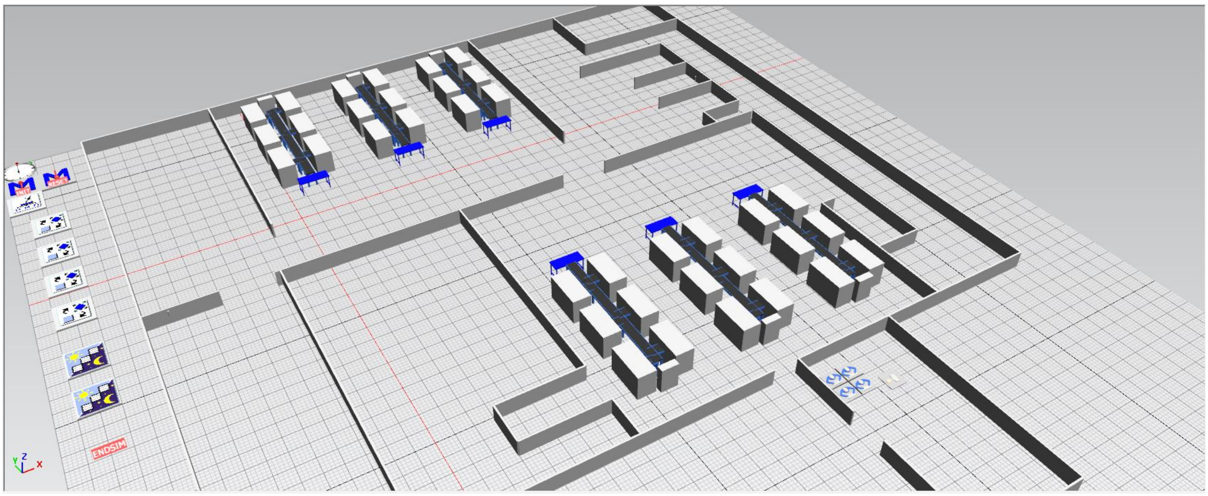


Figure 22: 3D perspective of the simulation model

Movable Units (MUs)

The simulation has four movable units, namely a tray (the finished product), vessel, box and starch bag. The trays are most used, since this is the product created by the production lines. They will be created in a machine and then moved using conveyer belts to the operator table where they will be packed in boxes. The vessels are created when a mixer is finished mixing a batch and will then be filled with 50 kilograms of material. They will then be brought to a machine or prefoamer by an operator. When they are empty, they will be brought the empty vessel drain and discarded.

Broker

All calls for requests for action, like the full vessels that have to be brought from the mixing machine to the production machines, are handled by a broker. This broker handles the requests on a first come first serve basis. The broker will ensure that operators that finish their task get a new assignment, if there are no requests open the operator will return to their workpool, where they will wait until the broker calls the operator for a new task.

Frames

The whole simulation is build up out of generic building blocks, called frames. These frames form the basis of the simulation model. These frames usually consists of methods, singleProcs, buffers, brokers and variables. A method is a piece of code that is executed when triggered. These triggers can be either another method or another event. SingleProcs are predefined processes, that last a predefined processing time. An in-depth description of these frames can be found in the Appendix Chapter 2.

Key methods

The simulation contains several methods to let everything work correctly. This section will highlight a few important methods.

StartSimulation
This method is triggered when the user presses the "Start Simulation"-button. The method resets all simulation specific variables and empties the tables. It then calls the runExperiment-method.

runExperiment
This method is called by StartSimulation, endSim and by itself. The goal of the method is to determine and start the next experiment. This method starts with checking if there are still experiments to run. If there are experiments to run, it will initialize some variables, like the current experiment number and if it is a optimization or not. For example the first experiment is always the original layout, in which the benchmark is created, so no new layout will be generated. If the current experiment is an optimization experiment, a new layout will be generated by calling the optimization-method. After a new layout is generated and a valid layout it found, all parameters and counters of the simulation are reset, all tables are emptied, the model is initialized again, so all starting parameters are set and the new location, if created, is drawn. The next simulation run is then started.

If the simulation ends, it will call the run experiment again, if then no more experiments needs to be run, the best found layout is redrawn and the simulation is stopped. We redraw the best found layout to show the user the layout that is advised.

optimizationMethod
This method is called by runExperiment. The goal of the method is to generate a new valid layout given the defined optimization-method. The method starts with initializing variables like the strategy and if needed which wall to follow. It then creates a list of attributes that have to be moved. After the list of attributes is created the method checks which strategy it has to follow. The first strategy is to follow a defined wall. Within this strategy the method loops through all earlier defined attributes that have to move. It than starts with placing the first attribute at the first possible location at the start of the wall (it is given if that has to be the west or east start point for the north and south wall or north/south for the west and east wall), the other attributes will be placed next to it with a given step size. This step size is a predefined distance, like 0.5 meter, that you move the attribute with. It checks if the attribute will be placed outside the building, if that is the case it will go back to the starting position, but now one step size off the wall, thus moving more inward into the building.

The second strategy is to generate random positions, within this strategy the method loops through all attributes to place. Per attribute it generates random x- and y-coordinates within the building. If the method places the same attribute at an invalid location 20 times it stops the current experiment.

generateNewPos
This method generates a new position for an attribute. The attribute it has to place is sent to the method when it is called. It is called by the optimizationMethod. It first determines what kind of strategy the current experiment has; this can be random or follow a wall. The exact method how layouts are created is explained in the previous chapter.

checkPosValid
This method checks if the just created position for the to be placed attribute is valid. It is called by the optimizationMethod. It starts by determining the west, north, east and south coordinates of the moved attribute. It then checks if the attribute overlaps a wall, it does this check by looping through all walls and check if the wall is horizontal or vertical. If the

wall is horizontal, the method checks if the wall is either north or south of the attribute, if this is not the case, it checks if the wall end west or starts east of the moved attribute, if this is the case the wall is not overlapping the moved attribute. If one of the above-mentioned checks is not true, then the wall is overlapping the attribute and the position is not valid. The method will then return false.

After the method successfully checked all the walls it will loop through all other attributes and checks if there is any overlap between the moved attribute and the attribute to check. This is done similar to that for the walls.

5.2 Simulation parameters

This section introduces the simulation specific parameters. It will explain the warmup length used, the run length of the simulation and the number of replications.

Warmup time

When the simulation starts, there are no moveable units (like vessels or produced products), so the whole model is empty. So it slowly has to fill with MUs, this means that in the beginning of the simulation the mixers are busier than normal and the operators that pack the finished product have nothing to do. This is not a representable scenario. That is why we use a warmup period, in which the simulation runs, but no results are collected. We start collecting data when the simulation reaches a steady state. Figure 23 shows the total amount of products created per shift in the Dutch facility. The graph shows that from the 4th shift the graph reaches a steady-state. We will use the 1st 3 shifts (1st day) as a warmup period and start collecting data at the 4th shift.

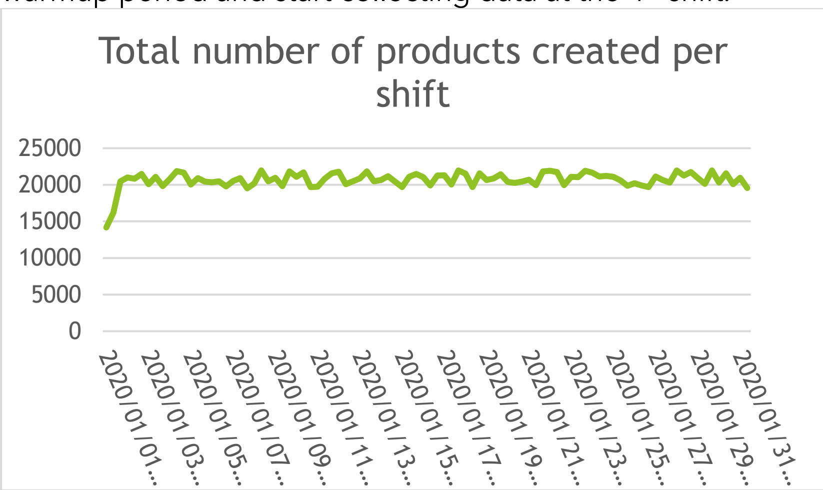


Figure 23: Total number of products created per shift in the simulated Dutch facility

Run length

The run length is the number of shifts that the simulation will run. As a rule of thumb you generally use 10 times the warmup length to get sufficient data. Since the warmup time is 3 shifts, or 1 working day, this would mean 10 days minimum run length. Since the production schedule could change per working day we decided to run the simulation for two weeks. Including the warmup time, this is a run length of 15 days.

Number of replications

To determine the number of replications that are needed to get a reliable average value, we should determine the maximum allowed error based on the confidence interval. We aim for a 95% confidence interval, which means that the allowed alpha is 0.05. The relative error is represented by: $\frac{\alpha}{1+\alpha}$. So the relative allowed error is 0.0476. We ran 30 replications. The number of replications for which the half of the confidence interval divided by the average is below the target allowed relative error, is sufficient. Figure 24 shows graphically the relative errors based on the confidence intervals per number of replications compared to the baseline.

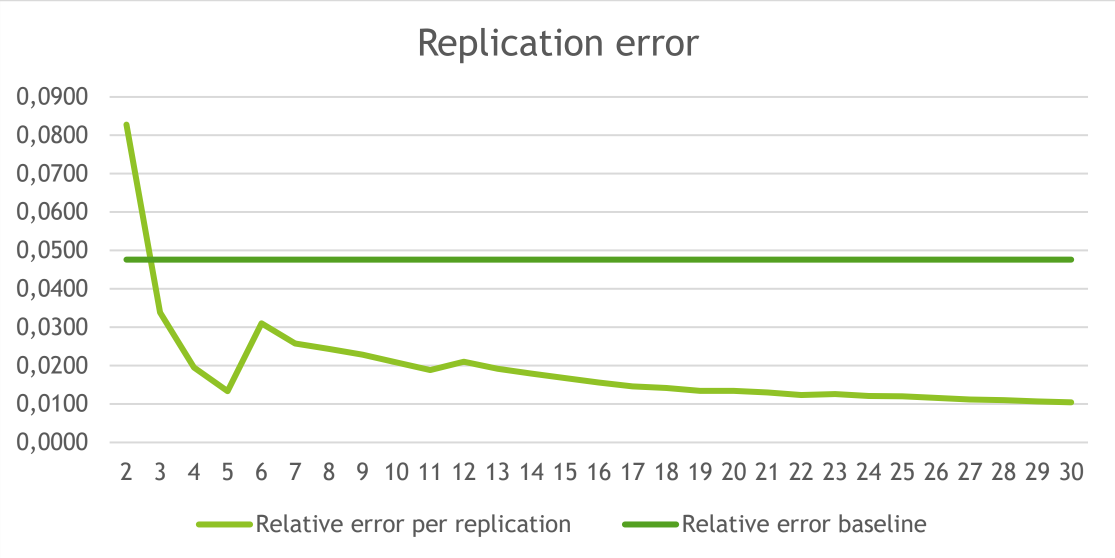


Figure 24: Relative error per number of replications

As can be seen in Figure 24 three replications is already sufficient to meet the allowed error. But since after that the error rises again, with a peak of 0.031 at 6 replications, we chose the earliest point where the error consistently drops off, which is at 6 replications. So for all experiments we will use 6 replications.

5.3 Conclusion

This section provides a conclusion to the software implementation chapter.

We have now developed a computer simulation model of the conceptual model introduced in the previous chapter. This simulation model both has 2D and 3D aspects to create obstacles for the operators to walk more realistic routes. Now we have created this model we should validate if the outcomes are realistic before we can draw conclusions.

6 Simulation validation

In this chapter we will discuss the validation and verification techniques we have used to determine the model's validity. Some of the techniques mentioned by Sargent (2010) are not suitable due to the lack of data, like comparisons to other models, since no similar computer simulation models existed yet.

Animation

One of the benefits of the chosen simulation software, Plant Simulation, is that it enables the option to graphically show the movement of the MUs in real time, both in 2D as in 3D. This feature is used in several steps of the model development. During programming of the model, we have used it to visually check if everything is moving as intended and without obstacles. After the model was finished the same feature was used to subjective validation with the client of this research, PaperFoam. We asked them if the movement shown in the simulation model corresponds to the reality, which they agreed with.

Face validity

Another technique we have used to validate our simulation model is face validity. In this stage we have asked experts in the field of computer simulations to take a look at some key methods and their output and verified with them that the results generated by these methods were conform expectations. For this validation step we have used employees of the company Ergo-Design, a company that is specialized in simulating real world production facilities.

Internal Validity

To determine the internal stochastic variability, we have replicated the experiment for the original layout of the Barneveld facility of PaperFoam a few times. For the validation we ran the production schedule as historical data, so there was no randomness here. During the experiments we use sampling to determine the production schedule, so we do have randomness. This aspect is mitigated using replications and then averaging the result. Between the experiments we found no statistically significant difference in total amount of products produced and total time that workers walked with vessels. This is as expected, since the production schedule for the whole experiment is fixed. A difference in outcome between the experiments would indicate that a variable is causing an unwanted negative effect on the outcome, like mixing times that are too long, which will entail that the machines have to wait for material.

Extreme-Condition tests

To determine if the simulation worked as expected we tweaked some input parameters to see if the simulation reacted as expected. We both used this during development of the simulation model, to see if the newly build module was working as expected and after finishing the simulation model to verify the whole model. One of the things we tested was setting the number of mixers to zero. This should result in no mixtures being completed and eventually all machines running out of material, so production would stop. This was exactly what happened.

Another parameter we changed was the reorder point for new vessels. We expected that if this reorder point would be too low, the machines did not order new material in time and had to wait longer to be restocked. As Figure 25 shows that a reordering point below 35% of the vessels capacity will negatively influence the total amount of products produced. This is happening because machines have to wait longer before new full vessels are brought to them. The graph also shows a smooth curve, indicating that there is indeed a relationship. This shows that the expected relationship between parts produced and availability of material on the machines works.

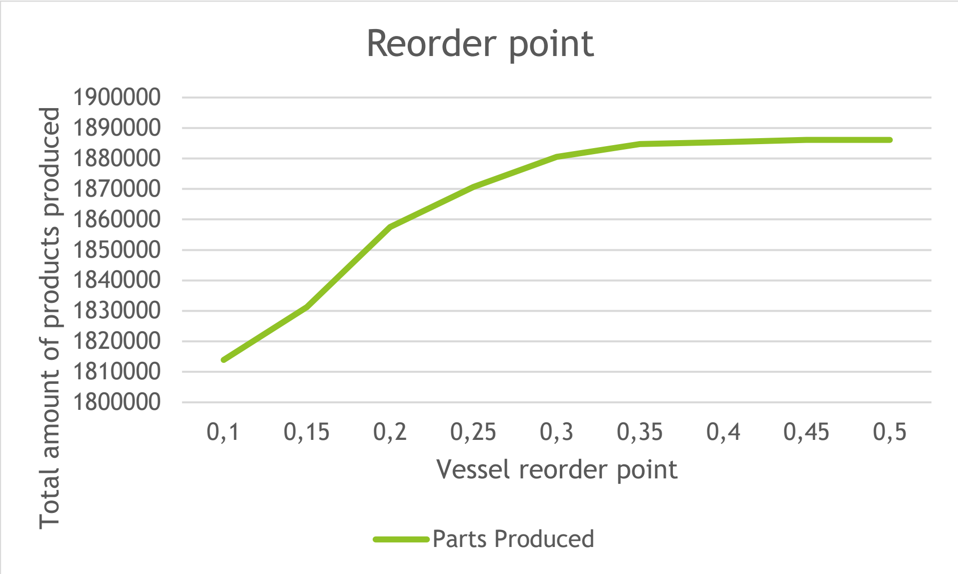


Figure 25: Total number of parts produced with respect to reordering point

Output verification

The final important verification step is to ensure that the output data generated by the model correctly represents reality. For this we compared the total number of products produced per day in the simulation model with the real number of products created on that day. For this validation tracing is used, which means that the production schedule is used as historical input data. So every shift the same products ran on the same machines as in the real world. Since the amount of product created heavily relies on the production schedule of that day, which is not constant over time. For example, not all machines ran every shift or day, resulting in highly fluctuating output. This meant a statical distribution over the data could not be determined, which meant that a standard T-test or chi-squared test could not be used.

A suitable statistical model that can cope with unknown statistical distributions is the Analysis of variance (ANOVA) method. ANOVA provides a statistical test whether two population means are equal. We set the null hypothesis as that there is no difference between the two datasets, in other words, the results from the simulated model are statistically equal to the real-world results. If the observed F-value is bigger than the critical F-value, we have to reject the null-hypothesis and cannot say that the simulation model gives similar results as the real world.

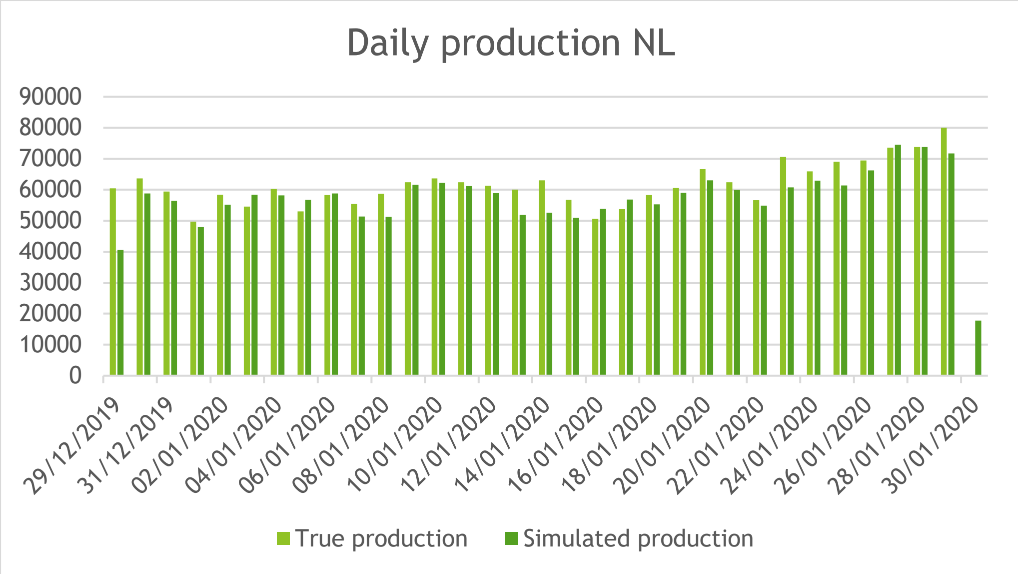


Figure 26: True and simulated daily production in The Netherlands

Figure 26 shows the simulated daily production presented next to the true daily production data for Dutch facility. The results of the other two locations can be found in the appendix. A few things are important to note that all three locations show in this comparison. On the first day the production of the simulated model is always a bit lower, this is why we ignore this using the warmup period of 1 day. The second thing to note is that in the simulated model we have some products that are created in the day after the simulation was supposed to end. This is because the simulation runs until the end of that shift, which is later than midnight, resulting in products being registered a day later. This is part of the cool down period of my simulation and will be ignored during further analysis. Figure 27 shows the results of the ANOVA analysis of the two datasets. As shown for a 95%-confidence interval the critical F-value is 4.0012. The observed F-value is 2.6656, resulting in us not being able to reject the null-hypothesis that there is no statistical difference between the two datasets. So, our conclusion is that with 95%-confidence we can state that the difference in output data between the simulation and reality is not significant.

ANOVA: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Column 1	31	1912442	61691,68	48956407
Column 2	31	1826289	58912,55	40867560

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	1,2E+08	1	1,2E+08	2,66555	0,10778	4,001191
Within Groups	2,69E+09	60	44911983			
Total	2,81E+09	61				

Figure 27: ANOVA results for The Netherlands

In this report we will only show the graph and ANOVA results of the Dutch facility. The other two can be found in the Appendix. Appendix Figure 4 shows the simulated daily production against the true daily production for Malaysia. Appendix Figure 5 shows the ANOVA results for Malaysia. The critical F-value for a 95%-confidence interval is 4.0069 and the observed F-value is 0.1155. So, with 95%-confidence we can state that we failed to reject the null-hypothesis that the simulated output data is significantly different than the real world data, so we can state that for Malaysia the simulation generates equal output data.

Appendix Figure 6 shows the simulated daily production against the true daily production for the US facility. Appendix Figure 7 shows the results of the ANOVA calculation. The critical F-value for a 95%-confidence interval is 3.9863 and the observed F-value is 1.1349. So with 95%-confidence we can state that also here we failed to reject the null-hypothesis that the simulated output data is significantly different than the real world data, so also here we can state that the model for the US generates equal output data as in the real world.

With the model passing the ANOVA test with all three locations and also the other validation tests we can state that all three models are representable for the real world and can be used to generate proven conclusions from.

Conclusion

The simulation model passed all validation tests, so we conclude that the simulation model gives accurate output data. This allows us to draw funded conclusions. We will now start to analyze the three existing locations.

7 Results

This chapter will present the results of created computer simulation model. First it will give some general results about the model. It then will give results from the layout optimization study for all three facilities of PaperFoam. Section 7.2 will give the results for Dutch facility in Barneveld. Section 7.3 will present the results for the Malay facility in Penang and section 7.4 will present the results for the US facility in Leland, North Carolina.

7.1 General simulation results

This section will present general results generated from the simulation model. First the effect of local search will be presented, then the experiments that were performed will be explained.

Local search

To investigate the effect of local search on the best found solution we ran the simulation on the Dutch facility with and without local search. Figure 28 shows graphically the difference. As can be seen Local Search does not always yield a better solution, for instance in experiment 9 there was no difference. Experiment 2 had a total walk time reduction of 43 minutes, or 11%. If local search results in a worse solution, this will of course not be saved.

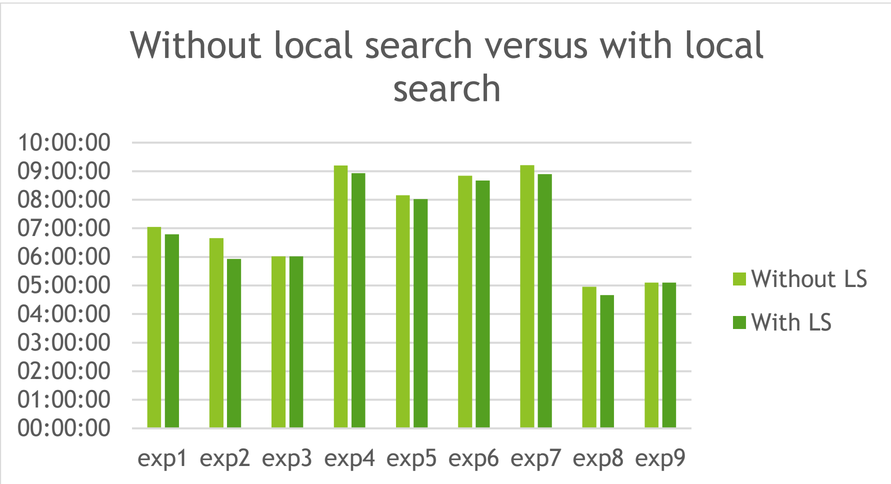


Figure 28: Graph showing the difference in walk times with and without local search for the same layout

Simulated path compared to straight path

To see the difference our created model would have compared to the existing models we compared the total walk time from our simulation with that of the existing model. For this we ran the original layout of the Dutch facility and tracked how many walks were done to each machine, and what the distance was that the operator had to walk to each machine. To get a reliable number of walks we ran the simulation with 6 replications and took the average of the number of walks required. Appendix Table 1 shows the distances measured in the simulation model and calculated using straight paths from the center of the mixers (straight through all objects) to the machines. On average the straight path distance is 18.5% shorter than the simulated distance, with extreme's up to 50% for machine 3.6.

To calculate the time that the operator would have walked if he would have used the straight paths, we used the same walking speed as in the simulation, 1.5 m/s. We then multiplied this by the distance and the number of walks the operator had to make to that machine. Appendix Table 2 shows the total walk times per machine for both the simulation (with simulated distances) as calculated for straight path distances. The total simulated walking time was 6 hours, 47 minutes and 26 seconds. The total estimated walking time using straight paths is 5 hours, 27 minutes and 13 seconds. So the total walk time using straight paths is 1 hours 20 minutes and 13 seconds less than the simulated time, which is an almost 20% underestimation of the total walk times. From this we conclude that using computer simulation we are able to calculate more realistic walking distances and thus getting more realistic required walking times.

Performed experiments

During testing of the simulation we found that the method that creates randomly generated coordinates for every attribute, we did not succeed to find a valid layout within 20 tries. This was mainly due to the fact that the attributes have a significant footprint, and thus always ended up overlapping.

We ended up running 9 experiments and within every experimented layout we performed local search to improve the walking times within that layout. Table 1 shows the experiments we ran per location. The 1st experiment is with the original layout. The next 2 experiments are following the north wall, first from west to east, then from east to west.

Exp. number	Strategy	Wall to follow	Start position
1	Original layout	n/a	n/a
2	Follow a wall	North	West
3	Follow a wall	North	East
4	Follow a wall	South	West
5	Follow a wall	South	East
6	Follow a wall	West	South
7	Follow a wall	West	North
8	Follow a wall	East	South
9	Follow a wall	East	North

Table 1: List of experiments that were performed

7.2 The Netherlands

This section will present the results from the simulation-based layout optimization study performed on the Dutch facility of PaperFoam. It will show the total walk times per experiment and the best found layout.

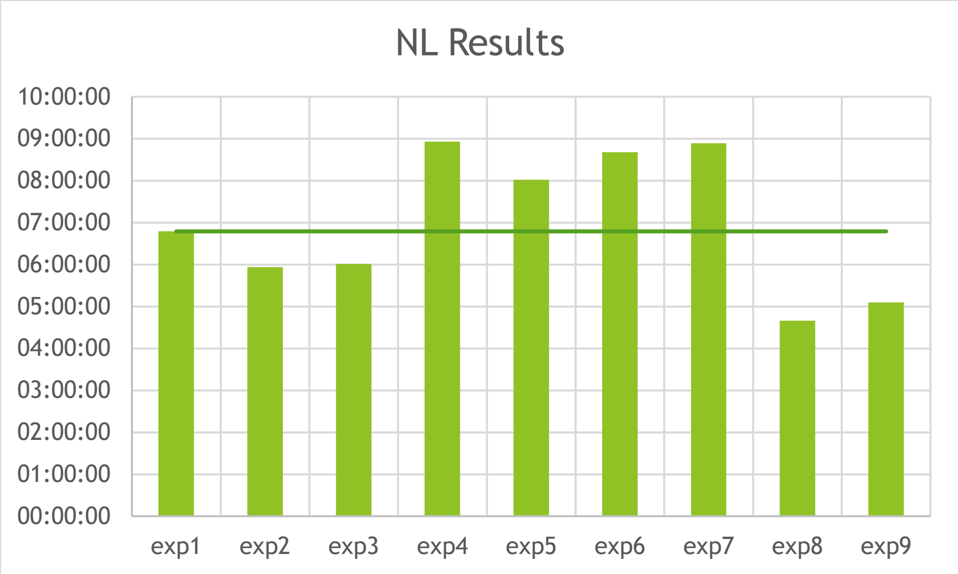


Figure 29: Graph of the total walk times in The Netherlands per experiment

Figure 29 shows graphically the summary of the 9 ran experiments. The first thing to note is the results of experiment 1. This is the original state of the Dutch facility. As mentioned before we ran the experiment for 15 days and did 6 replications per experiment. The total walking time to needed to run the original layout was 6 hours, 47 minutes and 26 seconds. The best-found layout was experiment 8, where the total walking time for the same schedule was 4 hours, 39 minutes and 53 seconds. A total reduction in walking time of 31%, or 2 hours, 7 minutes and 33 seconds. Figure 30 shows the best-found layout. As can be seen this follows the east wall, which is closest to the mixers, thus reducing the distance that needs to be traveled, so reducing the total walk time needed.

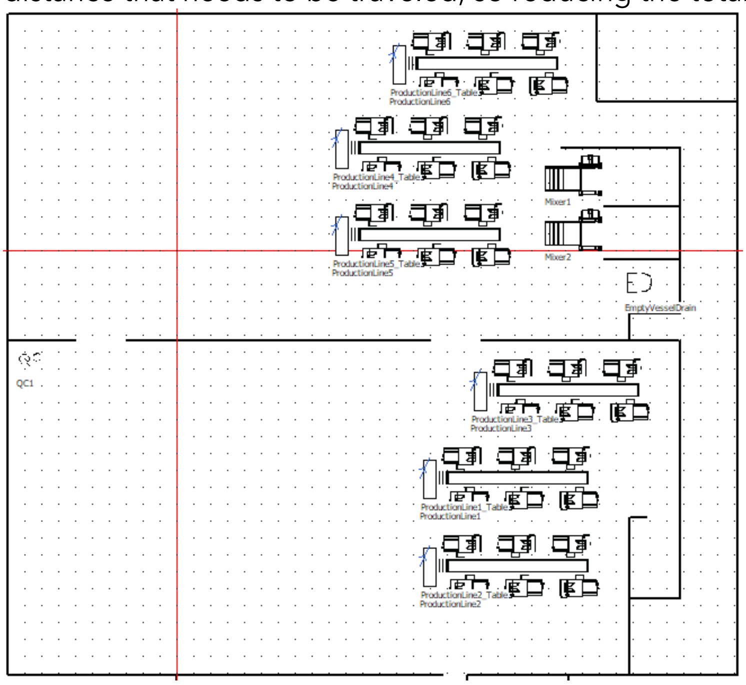


Figure 30: Most optimal layout for The Netherlands found

7.3 Malaysia

This section will present the results from the simulation-based layout optimization study performed on the Malaysian facility of PaperFoam. It will show the total walk times per experiment and the best found layout.

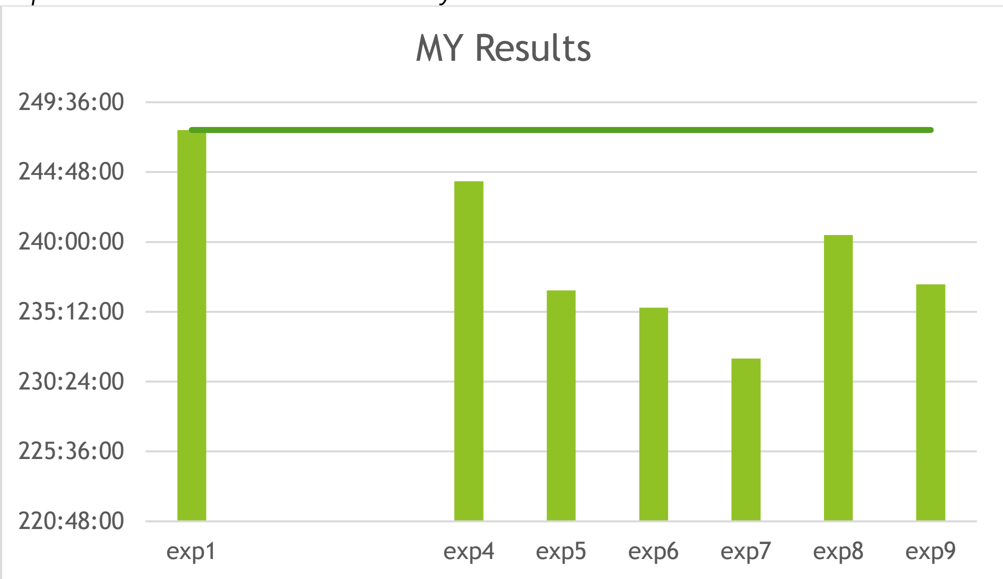


Figure 31: Graph of the total walk times in Malaysia per experiment

Figure 31 shows graphically the total amount of walking times per experiment. As can be seen there are no walking times registered for experiments 2 and 3. This is because of limited vertical room these two experiments did not yield feasible solutions. Experiment 1 is the original state. The total walking time was 10 days, 7 hours, 41 minutes and 33 seconds. The best-found layout was during experiment 7, where the total walking time was 9 days, 15 hours, 59 minutes and 50 seconds, a reduction of 15 hours, 41 minutes and 43 seconds. Although this is a small percentage reduction, namely 7%, it still is a decent absolute reduction. The main reason the reduction in total walking hours is less than for the other facilities is that a big part of the walking time is moving between the two buildings and floors. Figure 32 shows the layout for Malaysia generated in experiment 7.

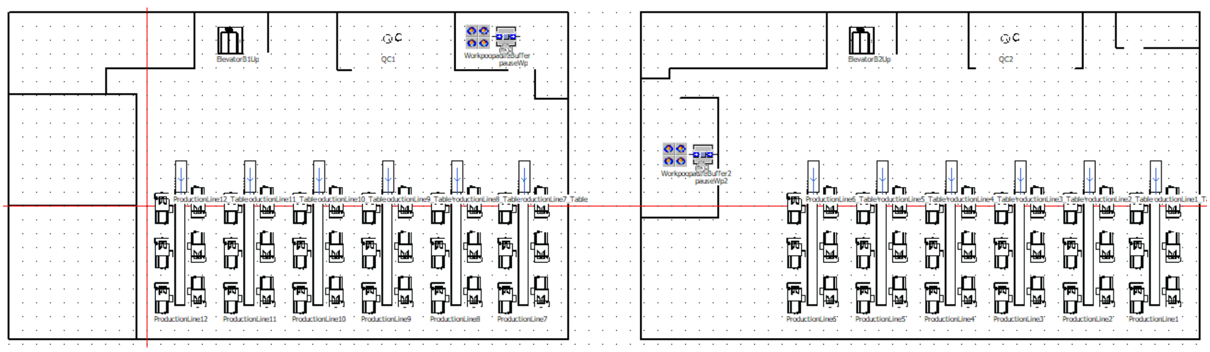


Figure 32: Most optimal layout for Malaysia found

7.4 The US

This section will present the results from the simulation-based layout optimization study performed on the American facility of PaperFoam. It will show the total walk times per experiment and the best found layout.

Figure 33 shows graphically the total amount of walking time per experiment. Experiment 1 is the original layout of the American facility. The total walking time was 9 hours, 58 minutes and 08 seconds. The best-found time was during experiment 6, where the total walking time was 5 hours, 13 minutes and 05 seconds. A total walking time reduction of 4 hours, 45 minutes and 03 seconds is achieved. This is a total reduction of 48%. Figure 34 shows the layout of experiment 6. As can be seen all production lines are way closer to the mixers, resulting in the big walking time reduction.

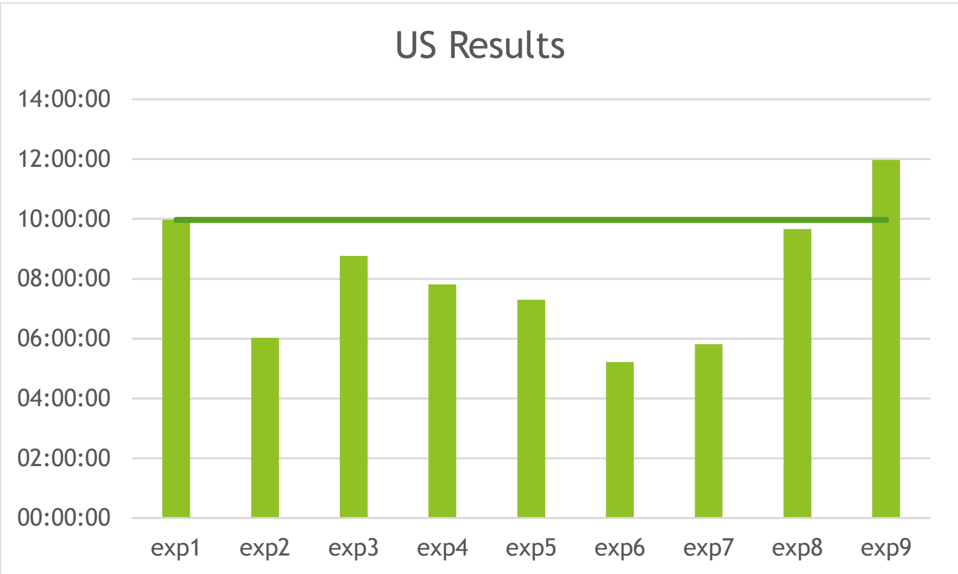


Figure 33: Graph of the total walk times in the US per experiment

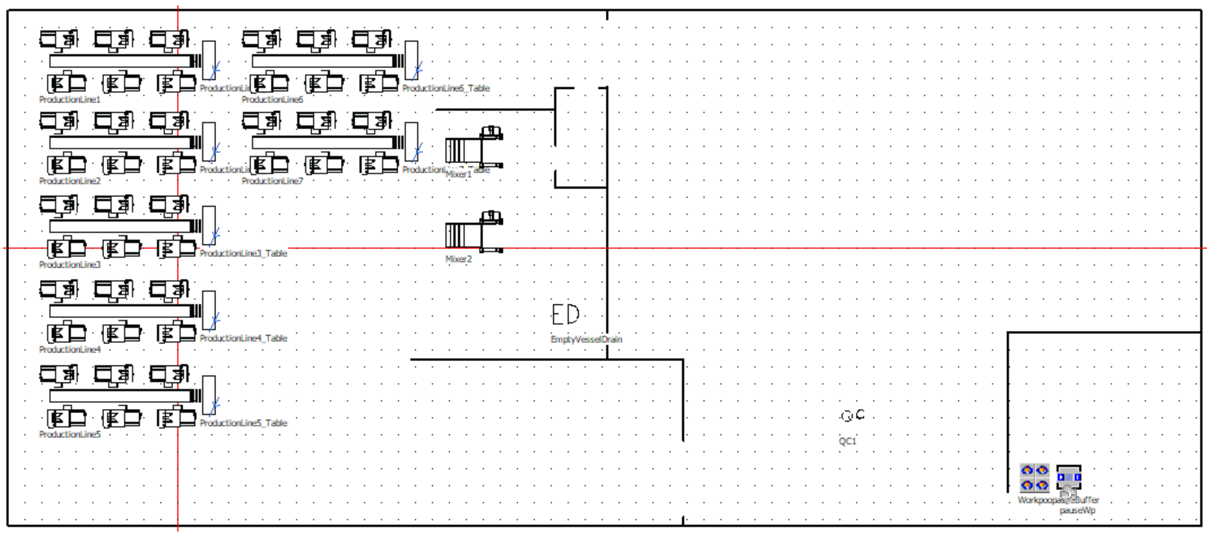


Figure 34: Most optimal layout for the US found

8 Conclusion

This chapter will conclude our research. Section 8.1 will give a brief summary of the steps taken and the final results found. Section 8.2 will set a critical note to our research. It discuss what could have been better or different in order to make the research more general. Section 8.3 will suggest possibilities for further research.

8.1 Conclusion

This section will conclude our research.

We started our research with a literature study where we discussed several well-known layout optimization models. We found that all models either need a lot of input data, which usually is not available, or use straight lines to calculate the walking time or distance between two objects. We found the solution for this in computer simulation. This solution can cope with limited input data and uses realistic distances between attributes. We introduced several steps needed to validate and verification computer simulations and have used these steps to later validate our simulation model.

We used a case study at PaperFoam's three production locations to identify the main waste in the interdepartmental flow. We also used these locations to gather input data like processing times for the different attributes, current layout and production schedule of a busy month.

After identification of the problem entity. We created a conceptual model to solve the layout optimization problem. We then translated this into a computer simulation model. This model was made in Tecnomatix Plant Simulation version 13. This model mainly consists of several building blocks and input data-frames.

When the simulation was validated, we could start with the optimization study. To generate new layouts, we used two strategies. A strategy where all production lines are placed along one of the four (north, east, south, west) walls, all in 1 line. The second strategy was to generate a random position per attribute to place. Within both strategies we checked if the generated layout was feasible, in other words no attribute overlapped another attribute or wall. After every experiment was ran, we used local search with simulated annealing to swap production lines around to see if this yielded improved layouts. During this phase we found that randomly generated layouts did not create feasible solutions, so we only used the strategy to follow the walls. Within every experiment we ran a local search to swap production lines within the same building among each other depending on the found walking times per line. This swap was done between the production line with the longest and shortest walking times and the second longest and shortest. Determining if a solution was accepted was done using simulated annealing.

We compared the same layouts using local search and without local search. We found that a maximum total walk time reduction of 11% could be reached by only using local search. A simulation run of 15 days took, depending on the location, around 1 minute. So running it including the replications meant that it took 6 minutes to simulate a generated layout. The time to run a simulation run is highly dependent on the computer CPU.

To see the difference in expected walk distance and total walk times for the operators we compared the simulated distance with the distance calculated using straight paths. From this we found that the straight path distance is on average 18.5% shorter from the mixers to the production machines. With an extreme of 50% shorter for machine 3.6. This resulted in a difference in expected walking time. Using our simulation we found an expected walking time of 6 hours, 47 minutes and 26 seconds, while we calculated an expected walking time of 5 hours, 27 minutes and 13 seconds if we would have used straight paths. This is almost a 20% underestimation of the total walking time. Thus showing that using a computer simulation to determine realistic distances gives a more realistic representation.

For the Dutch facility in Barneveld we found a solution that has a 31% reduction in total walking time needed for the month January. This was an absolute reduction 2 hours, 7 minutes and 33 seconds. For the Malay facility in Penang we found a smaller reduction, namely 6% and an absolute reduction of 15 hours, 41 minutes and 43 seconds. The smaller percentage has as a main reason that the Malay facility consists of two separate buildings. Furthermore, the mixers are on another floor than the production machines, this results in the majority of the movement happening between the buildings, where we did no further optimization in. For the US facility in Leland, North Carolina we found a total walking time reduction of 48%, this is an absolute reduction of 4 hours, 52 minutes and 54 seconds. This is a major saving. This is mainly due to the fact that the production lines were placed directly next to the mixers.

8.2 Discussion

This section will set a critical note to our research. We discuss what could have been better or different in order to make the research more general.

The developed simulation model only takes one variable, namely the interdepartmental flow from the mixer to the production machines into account. Within an optimal layout there are more factors that have a role to play, like other material flows or technical challenges, like dust production of the mixing process or warehouse space. This would have resulted in additional constraints where certain attributes could not be placed, thus limiting the amount of possible outcomes.

Furthermore, we used several assumptions which have an impact on the performance of the simulation. For example assuming that the machines never break down and are available 100% of the time. We also assume that the walkers walk with a constant speed and walk the most efficient path from the mixer to their destination.

Within our research we only focused on optimizing the layout of the production lines. The other attributes like the mixing machines, quality control and empty vessel drain remained unmoved. We focused only on the moving of the production lines because we identified the material flow between the mixing machines and production lines as being the most wasteful. Since it technically not easy to move the mixing machines, we only focused on the production lines. We also kept the original outer walls of all three facilities and kept the same number of production lines within the original building.

8.3 Further research

This section will give options for further research that will broaden the reach of our research.

The current simulation model is only tested and verified for one case study, namely for PaperFoam. In order to proof its broader use in the field of production facility layout optimization it should also be tested for other industries or companies.

Furthermore, within our research we only focused on one movement flow, namely of the raw substance vessels from the mixers to the production machines or prefoamers. We have not looked at other movement flows like the incoming raw material and of the finished products from the operator tables to the boxes and then of the full boxes to the warehouse.

Our research also uses a given production schedule for a given month. The simulation could be used to generate a new schedule that will optimize the required walking hours whilst ensuring the same output. For the client this could have resulted in a cheaper and easier to implement solution. Think about utilizing the machines closer to the mixers more than the other machines and giving priority to products that need a lot of material.

9 References

Arya, V., Garg, N., Khandakar, R., & Meyerson, A. (2004). Local search heuristics for k-median and facility location problems. *Society for Industrial and Applied Mathematics*, 544-562.

Balakrishnan, J., & Cheng, C. H. (1998). Dynamic layout algorithms: A state-of-the-art survey. *Omega, 26*(4), 507-521.

Baykasoglu, A., & Gindy, N. N. (2001). A simulated annealing algorithm for dynamic layout problem. *Computers & Operations Research, 28*(14), 1403-1426.

Braglia, M., Zanoni, S., & Zavanella, L. (2003). Layout design in dynamic environments: Strategies and quantitative indices. *International Journal of Production Research, 41*(5), 995-1016.

Chhajed, D., Montreuil, B., & Lowe, T. J. (1992). Flow network design for manufacturing systems layout. *European Journal of Operational Research*, 145-161.

Chiang, W.-C., & Chiang, C. (1998). Intelligent local search strategies for solving facility layout problems with quadratic assignment problem formulation. *European Journal of Operation Research*, 457-488.

Das, S. K. (n.d.). A facility layout method for flexible manufacturing systems. *International Journal of Production Research, 31*(2), 279-297.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility Layout Problems: A survey. *Annual Reviews in Control 31*, 255-267.

Foulds, L. R. (1983). Techniques for facilities layout: Deciding which pairs of activities should be adjacent. *Management Science, 29*, 1414-1426.

Francis, R. L., & White, J. A. (1974). *Facility Layout and Location: An Analytical Approach.* Englewood Cliffs: Prentice-Hall.

Fruggiero, F., Lambiase, A., & Negri, F. (2006). Design and optimization of a faciliity layout problem in virtual environment. *Proceeding of ICAD 2006*, (pp. 2206-2218).

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 671-680.

Kleijnen, J. P. (1995). Verification and validation of simulation models. *European Journal of Operational Research, 82*, 145-162.

Koopmans, T. C., & Beckmann, M. (1957). Assignment Problems and the Location of Economic Activities. *Econometrica, 25*, 53-76.

Law, A. M. (2015). *Simulation Modeling and Analysis.* Ashford Colour Press Ltd.

Martis, M. S. (2006). Validation of Simulation Based Models: A Theoretical Outlook. *The Electronic Journal of Business, 4*, 39-46.

Mavridou, D. T., & Pardalos, M. P. (1997). Simulated Annealing and Genetic Algorithms for the Facility Layout Problem: A survey. *Computational Optimization and Applications*, 111-126.

Meller, R. D., & Gau, K. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems, 15*(5), 351-366.

Mes, M. (2017). *Simulation Modelling using Practical Examples.* Enschede: University of Twente.

Muther, R. (1961). *Systematic layout planning.* Management & Industrial Research Publications.

O'Brien, C., & Abdul Barr, S. E. (1980). An interactive approach to computer aided facility layout. *International Journal of Production Research, 18*, 201-211.

Sankey, M. H. (1896). The Thermal Efficiency of Steam-Engines. *Minutes of Proceedings of the Institution of Civil Engineers*, 182-242.

Sargent, R. G. (1994). Verification and Validation of Simulation models. *Proceedings of the 1994 Winter Simulation*, 77-87.

Schmidt, M. (2008). The Sankey Diagram in Energy and Material Flow Management. *Journal of Industrial Ecology, 12*(1), 82-94.
Shannon, R. E. (1975). *Systems simulation: the art and science.* Prentice Hall.
Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. (2010). *Facilities Planning*. New York: Wiley.

Appendix

Table of contents

1. Sargent's validation techniques

Sargent (1994) presents different verification and validation techniques. The techniques can either be subjective or objective, usually a combination of these are used. Some validation techniques are:

- *Animation (Operational Graphics),* this is a subjective verification technique where the model's behavior is shown graphically, an example is parts moving through a factory. The user can than declare if everything he sees is according to his expectations.
- *Comparison to Other Models,* this is an objective verification technique where various model results are compared with results from other validated models.
- *Degenerate tests,* this is an objective verification technique where the degeneracy of the model's behavior is tested. This is achieved by removing portions of the model or changing values of input and/or internal parameters. An example of this if investigating if the average number of customers in a queue of a single server continues to increase when the arrival rate is larger than the service rate.
- *Event Validity,* this is a subjective verification technique where the events occurrences of the simulation model are compared to those of the real system to determine if they are the same.
- *Extreme-Condition Tests,* this is an objective verification technique where the model structure and output should be plausible for any extreme and unlikely combination of factors in the system. An example for this is that the production output should be zero if the inventory of required material is zero.
- *Face Validity,* this is a subjective verification technique where people with knowledge about the system or process are asked whether the model and its behavior are reasonable.
- *Fixed Values,* this is an objective verification technique where all input and internal variables are fixed, thus making it possible to check the results against the easily calculated expected results.
- *Historical Data Validation,* this is an objective verification technique where, if it exists, historical data is used. Part of the data is used to build the model; the remaining data is used to determine if the model behaves as the real system does.
- *Internal Validity,* this is an objective verification technique where several replications of a stochastic model are made to determine the amount of internal stochastic variability. A high amount of variability may cause the model's results to be questionable.
- *Parameter Variability (Sensitivity Analysis),* is an objective verification technique where the effect on the model's behavior and its output are determined by

constantly changing the input and internal parameters of a model. The same relationship should occur in the model as in the real system.
- *Turing test,* Schruben (1980) introduced this method. This is a subjective verification technique where results of either the simulated method or the real system are presented to people with knowledge about the system, they are then asked to identify which results come from which systems output (simulated or real).

2. Additional details for simulation model

Figure 35 shows a screenshot of Plant Simulation. On the top there are the simulation controls, on the left the different frames and just above the frame itself the toolbox.

The factory-frame

The first frame is the factory itself. This is the main frame where the simulation runs in. This frame contains the following attributes:

- Brokers, one for the maximum amount of buildings found (in our case 4).
  - Brokers handle the request from stations for operators to go there.
    - Every broker will handle its own building.
- Two shift calendars
  - ShiftCalendar, this contains the current shifts the factory is working on, including breaks and the days.
  - notWorking, whenever the factory is closed this shift calendar is activated, which results in none of the workers leaving the workerpool.

The frame also has the following three methods:

- MainInit
  - This method is triggered when a simulation is initialized
- Reset
  - This method is triggered the simulation is reset.
- endSim
  - This method is triggered when a simulation run ends.

The layout of the factory, so the walls and the placement of all attributes, are drawn every time the simulation is started. Besides the production lines and mixers there are also the following attributes:

- Quality Control (QC)
  - This is the place where all full boxes are brought to.
- Empty vessel drains
  - This is the place where all empty vessels coming from the machines (or prefoamer) are going to. Normally they will be cleaned and put back into rotation, but this simulation generates a new vessel everytime a mixer is done with its batch.
- workerPool
  - This is the place where all walkers are sent from. If they do not have any work or when they go on a break they will return to this spot.
- pauseBuffer
  - This buffer will be used by walkers who are carrying something at the start of the break, they will drop-off what they have at the start of the break and pick it up after the break.

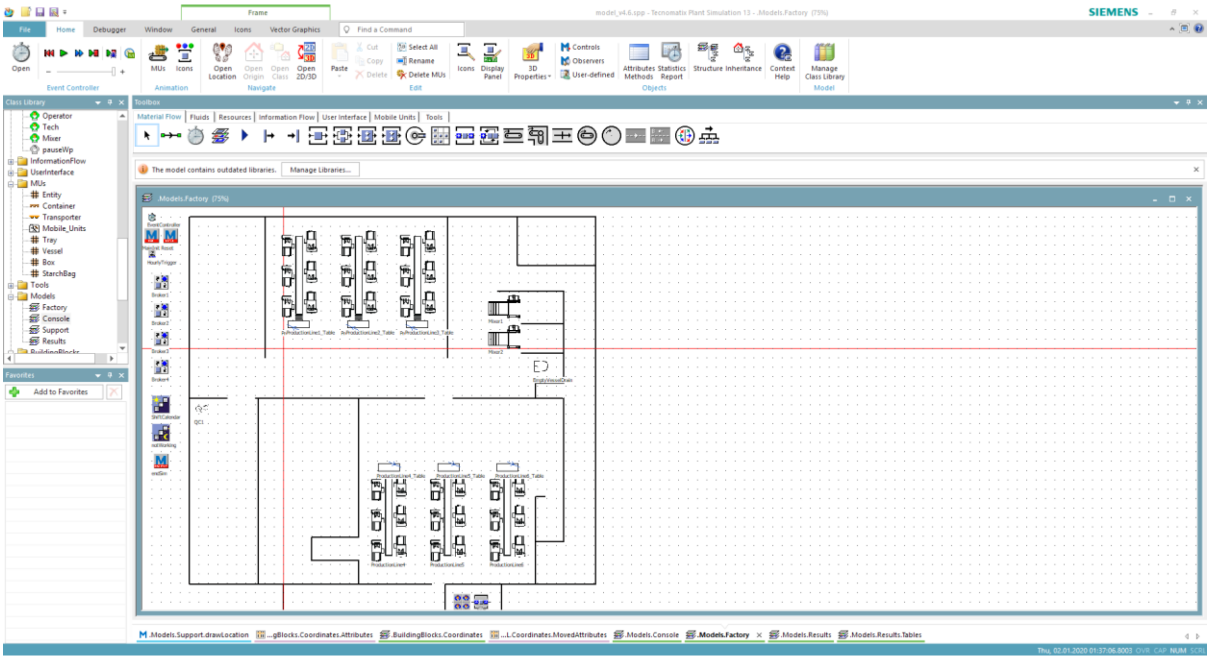


Figure 35: Screenshot of Plant Simulation with the factory frame

The console-frame
This frame is for the user to use to set up the simulation. It has a drop-down menu to select the site the user wants to run and a button to start the simulation. Furthermore, the frame has a lot of variables that can be changed by the user to change the simulation, these variables are categorized by:

- Start settings
    - These settings determine when the simulation starts, if unchanged it will start at the beginning of the given production schedule. If changed the simulation will start at the defined date and shift.
- Mixing settings
    - These settings determine how the mixing machines react, for instance the minimum batch size determines how many vessel orders of the same recipe have to be ordered before they start mixing (a vessel contains around 50kg of material, so 100kg minimum means two orders have to be placed).
- Destination settings
    - These settings determine where the full boxes and empty vessels go to.
- Optimization settings
    - These settings are important for the simulation optimization. The table experiments contain the different experiments that will be ran. These experiments can be defined by the user.
- Visual settings
    - These settings determine how the simulation will be shown.

Figure 36 shows a screenshot of this frame.

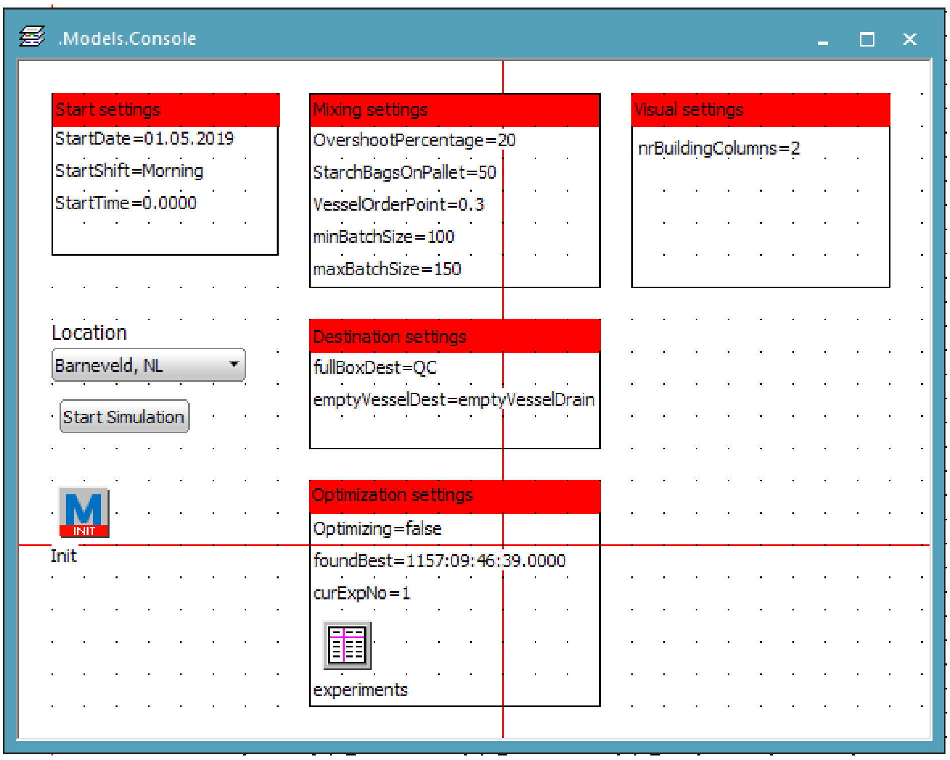


Figure 36: Console frame

Data frame

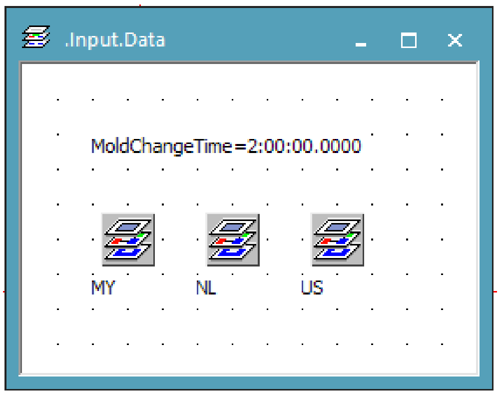


Figure 37: Screenshot of general data-frame

The specific location information is collected in the location information-frame, shown in Figure 38. It has five different categories of information. The first category hold location information, this has a table with location information like the amount of buildings and a frame that has tables with the coordinates of the walls and current attributes. The next category contains the schedule information of that location, this is information like the production schedule, the operator schedule and the shifts that the facility runs. The next category contains specific variables like the number of mixers and the maximum number of workers, if relevant. The next category contains the facility specific times like the mixing times and operator handling times. The last category contains the relationships between buildings, like what door or elevator goes from which building to another building. Figure 37 shows the data-frame that contains all locations in the simulation.

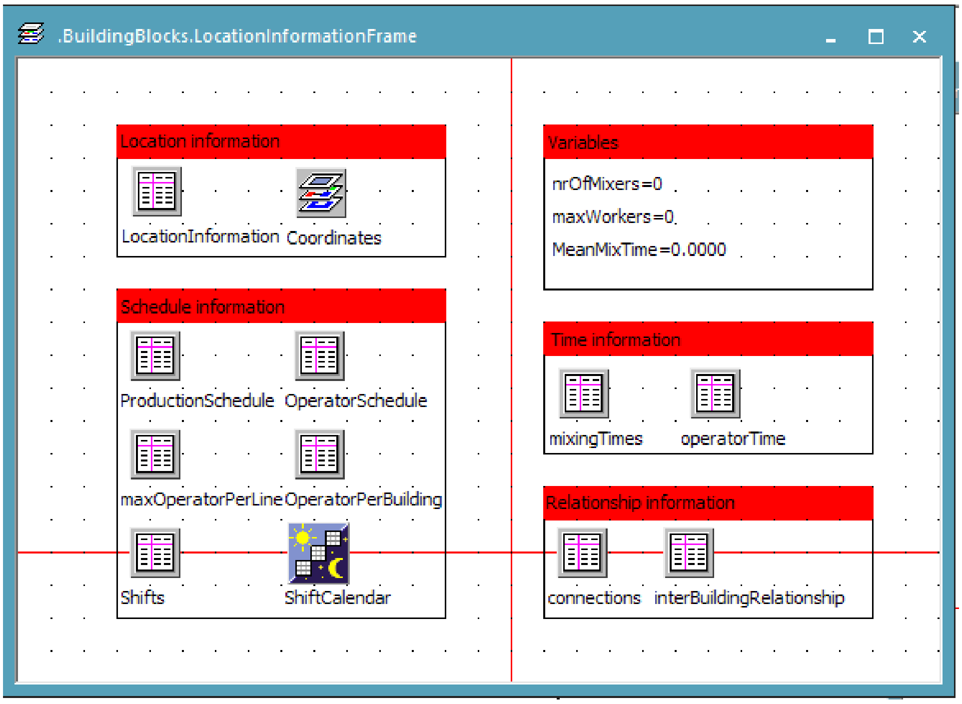


Figure 38: Screenshot of location information-frame

Building blocks

The simulation contains serval generic building blocks. This section will introduce and explain them. First, we will explain the most used building block, namely the production lines, this building block contains another two building blocks, namely a production machine and the prefoamer. Furthermore, there are three other building blocks we will introduce, namely the mixer, quality control and the empty vessel drain.

Machine

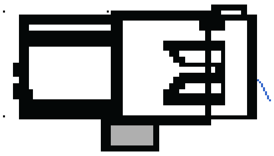


Figure 39: Icon of the machine

Figure 39 shows the icon that is visible when you place the machine in another frame. Figure 40 shows the frame of the production machine. On the left there is a list of machines setting variables, these settings will hold information like if the machine is currently running or not and if the machine is running, which product it is currently being created and the production specifications of that product. On the top there are ten methods that keep the machine running, among those methods is an initialization-method that defines the starting settings of the machine, two methods that get the product info and the material info for the current product that has to be created. These methods are called every start of a new shift. The other methods arrange the start of production, including the removal of material and if needed the ordering of new vessels. There are three machine variables that keep track of the machine during the simulation, like the total amount of products created and the time the last production was created. Under that there are two important blocks. The production simulation block resembles the machine producing. A MU is waiting in the buffer, when the machine starts production (with the StartProduction-method) the MU is moved to the MachineBuffer, it should then automatically enter the SingleProc. The process time of the SingleProc is determined by the CycleTime-variable. At the end of the process time the FinishProduction-method is called. This method puts the MU back to the Buffer and creates the by Cavities-variable defined number of products in the NewProdBuffer. A similar method is used to simulate the mold change of a machine. While a mold change is in process, the machine cannot produce any products.

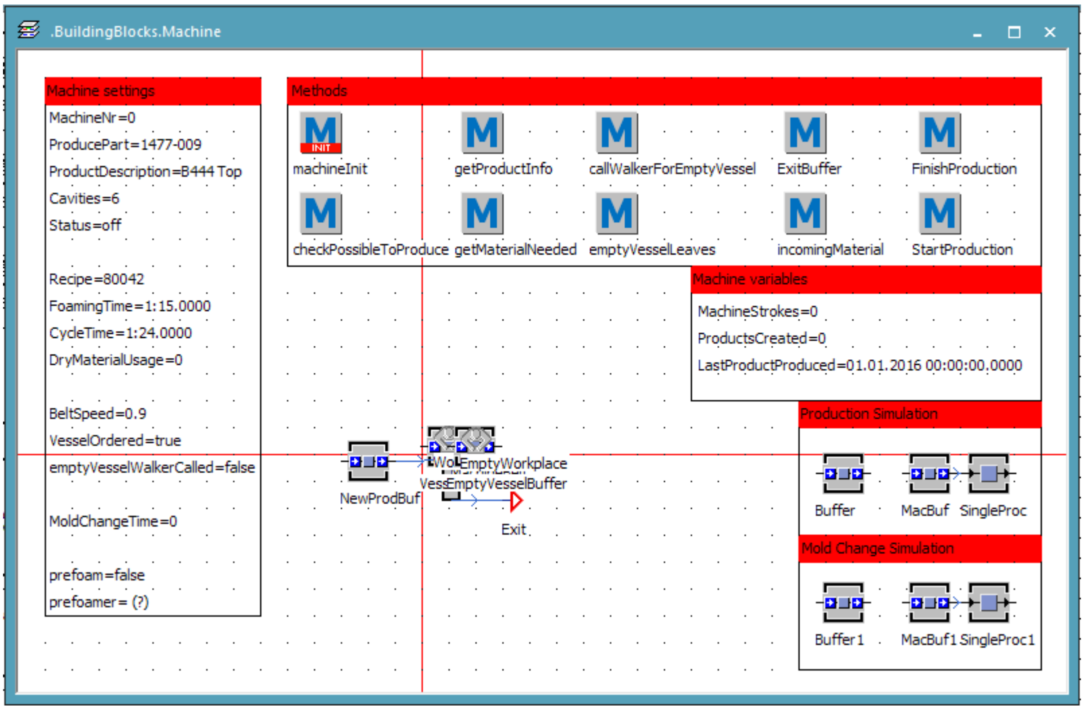


Figure 40: Production machine frame

Prefoamer

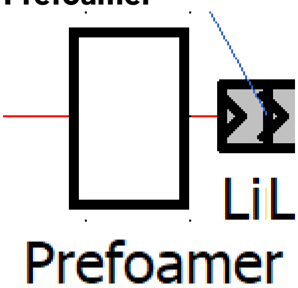


Figure 41: Icon the prefoamer

Figure 41 shows the icon of the prefoamer, this is building block is usually used within the production line-frame. Figure 42 shows the frame of the prefoamer. The prefoamer has six setting-variables. The Used-boolean shows if the Prefoamer is being used, if true the vessels with the recipe defined in the Recipe-variable will be brought to this prefoamer instead of the machines using that recipe in the production line. The seven methods run the prefoamer. It has checks to see if there is enough material still left in the vessels to run another production run in one of the attached machines, a method that removes the required amount of material from the vessel and a method that checks if a new vessel should be ordered. A new vessel is ordered when the amount of remaining material in a vessel drop under the reordering point. The reordering point is a in the console defined percentage-variable.

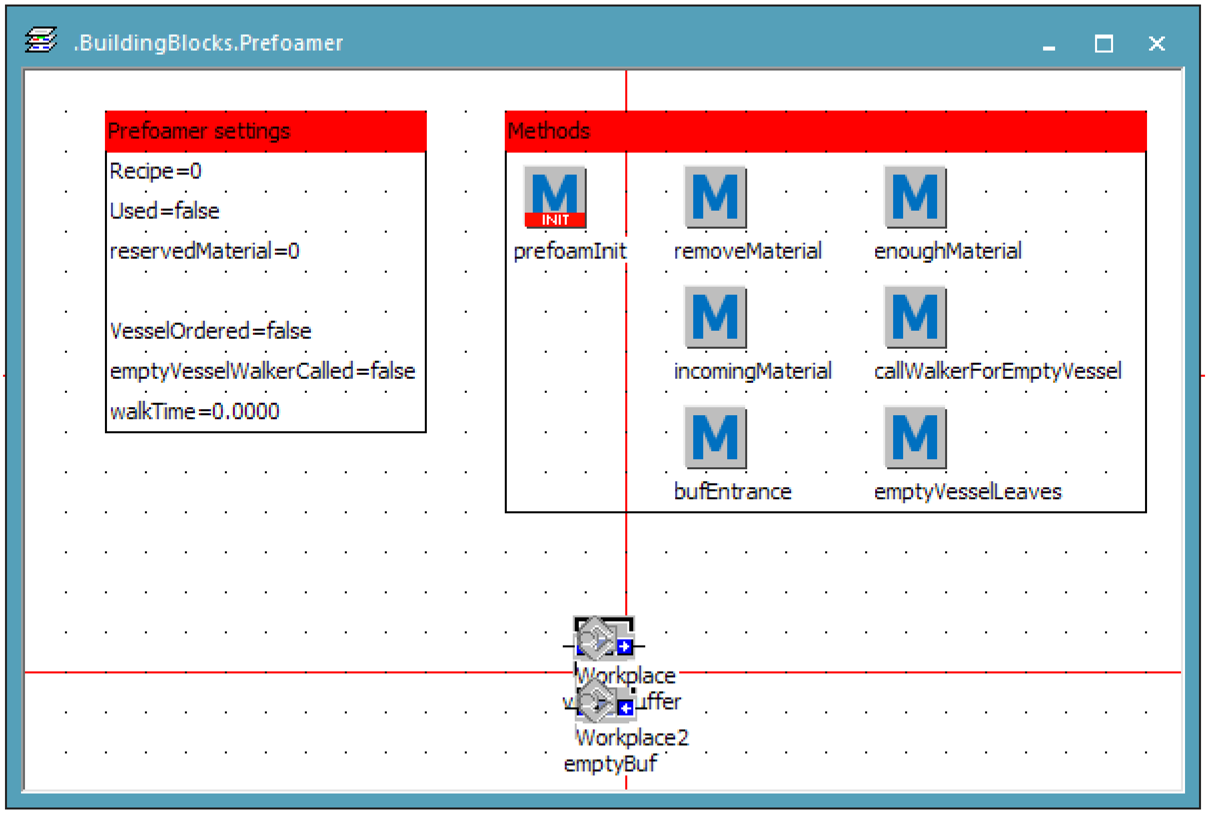


Figure 42: Screenshot of prefoamer-frame

Production line

Figure 43 shows the icon of the production line-frame. Every production line consists of several production machines, several conveyer belts and a prefoamer. Figure 44 shows the frame of the production line. It has three variables, namely LineNr, broker and totWalkTime. LineNr will show the line-number of that production line, this will make it easy to match the production schedule with the right production line. Broker defines which broker handles the demand of the machines; this broker is depending on the building. totWalkTime will keep track of the total walking time that was needed to serve the production line. Furthermore, the frame has a table file with all information about the machines, like if it is running or not, number of products produced, what product it currently is producing and its statistics like, running-, waiting- and blocked-percentage.

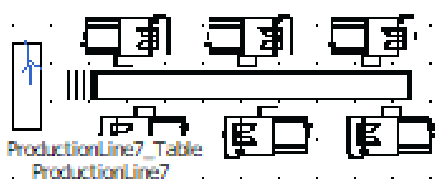


Figure 43: Icon of the production line

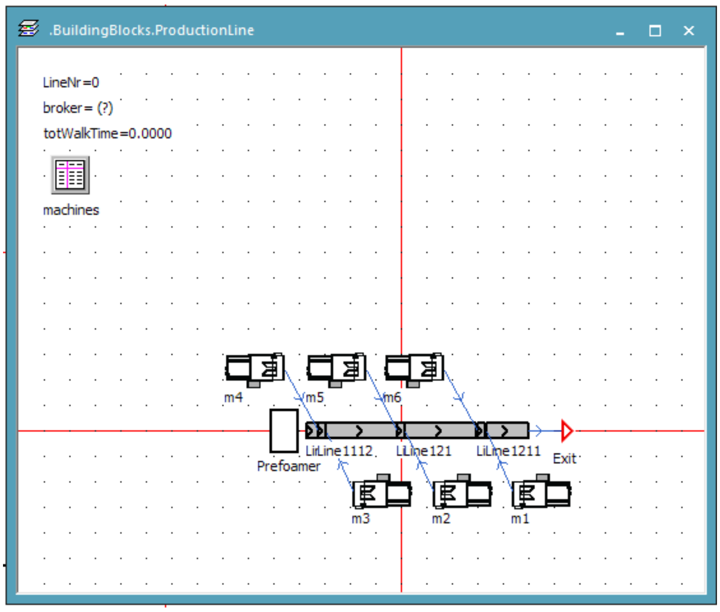


Figure 44: Production line frame

Mixer

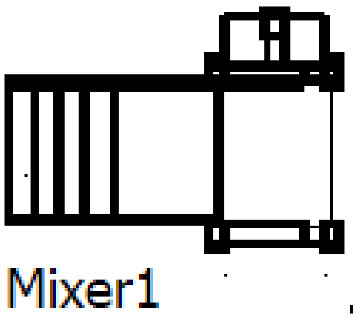


Figure 45: Icon of the mixer

Figure 45 shows the icon of the mixer that will be placed within the Factory-frame. Figure 46 shows the mixer-frame. The frame consists seven Setting-variables. The status-variable declares the state of the mixer, this can be running, stopped or blocked. It further shows, via the LastProduced-variable, when the last batch was made. orderToMake-variable is a list with the orders that the mixer is currently producing, this is used to find the destination of the created vessels. The mixing-simulation block works similar to that of the production machine. A MU is waiting in the MixBuf until it is moved to the Buffer by the StartProduction-method. The singleProc's processing time is defined by the MixingTime-variable. The value is randomly drawn from the MixingTimes-table in the Dataframe. In the EndProduction-method the full vessels are created, the number of vessels is depending on the number of orders that the mixer is currently producing. These vessels will be placed in their own vesselBuffers and mixers will be called to pick up these vessels. The checkPossibleToMix-method checks if the mixer currently can start mixing, this is depending on several factors. First of all, the mixer should be empty and currently not mixing another batch. Further the mixer should have enough raw material to create a new batch, in the current simulation the stock of raw material is set to infinite, so this is always true. The method also checks if there is enough demand from the same recipe to fulfill the in the console defined minimum order quantity.

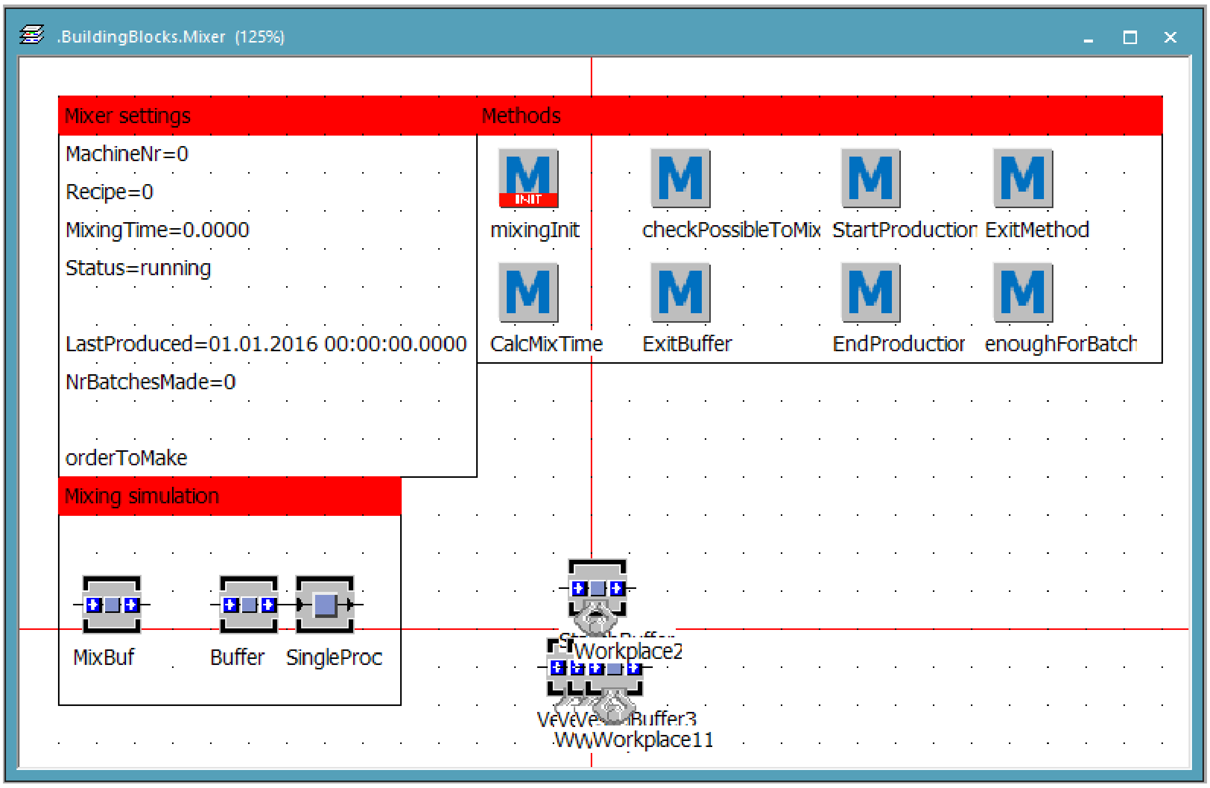


Figure 46: Screenshot of the mixer-frame

Operator table

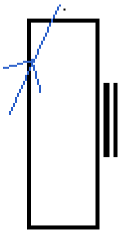


Figure 47: Icon of the operator table

Figure 47 shows the icon of the operator table, which is connected to the exit-connector of the production line-frame. Figure 48 shows the Operator Table-frame. The method contains seven methods controlling the operators. It further contains a variable defining the number of operators working on this table. The settings per operator are grouped in the wpData-table. Where the processing time, the state (working or not working) and the last worked time are stored. All products coming from the production line enter the frame in through the Entrance-connector and will then be moved into the incomingBuffer. Every free operator will keep checking if there is a product waiting for them in the IncomingBuffer, if this is the case, the product-MU will be moved to that operatorStation. A processing time will be determined using the given distribution in the data frame. At the end of the process the finishOperation-method will be called. Within this method the integer holding the number op products in the box will be increased and the method checks if the box is full. If this is the case that box will be moved to the fullBoxBuffer. A walker will then bring that box to the Quality Control-attribute.

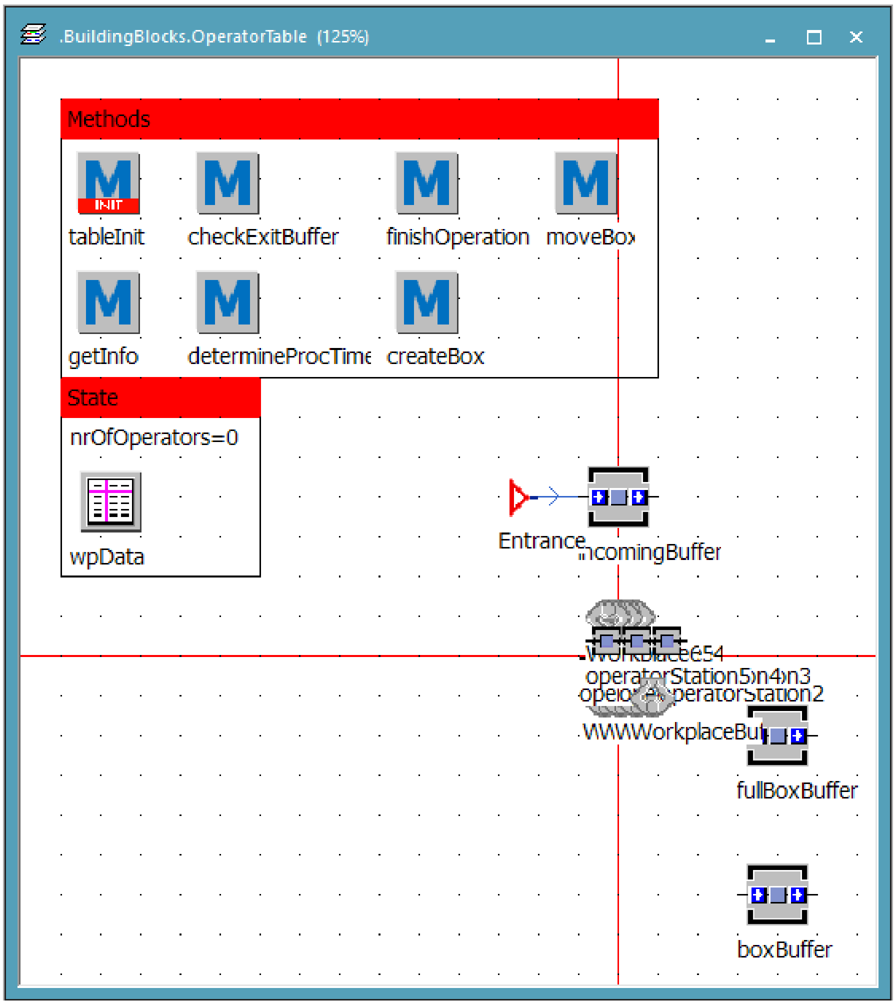
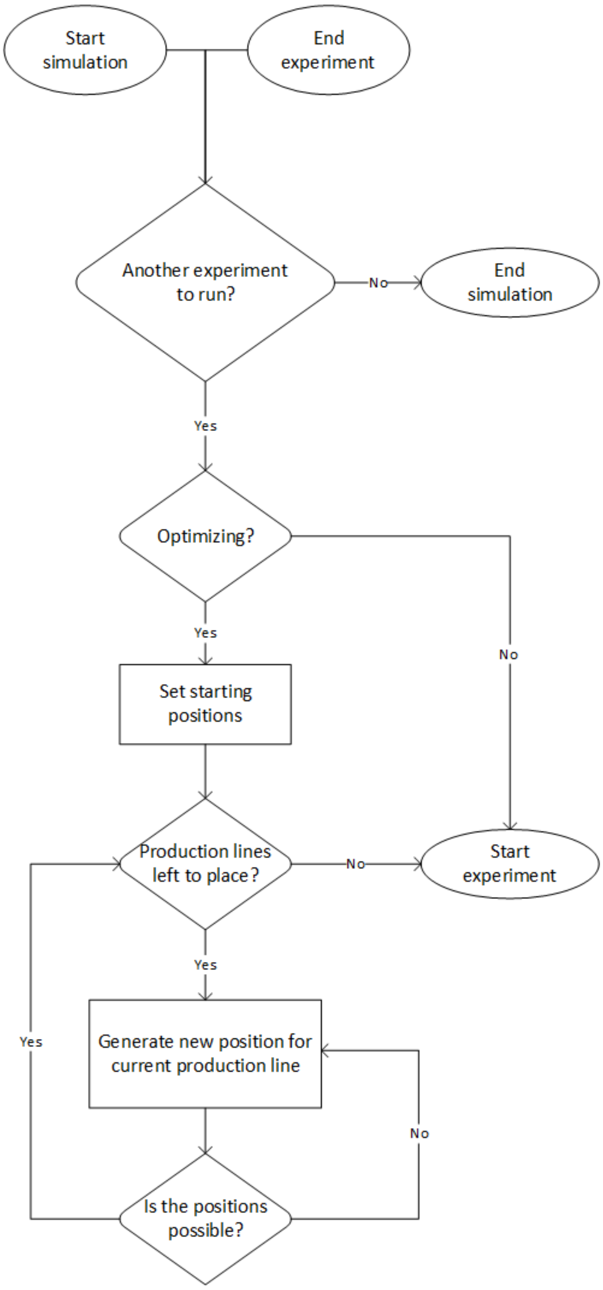


Figure 48: Screenshot of Operator Table-frame

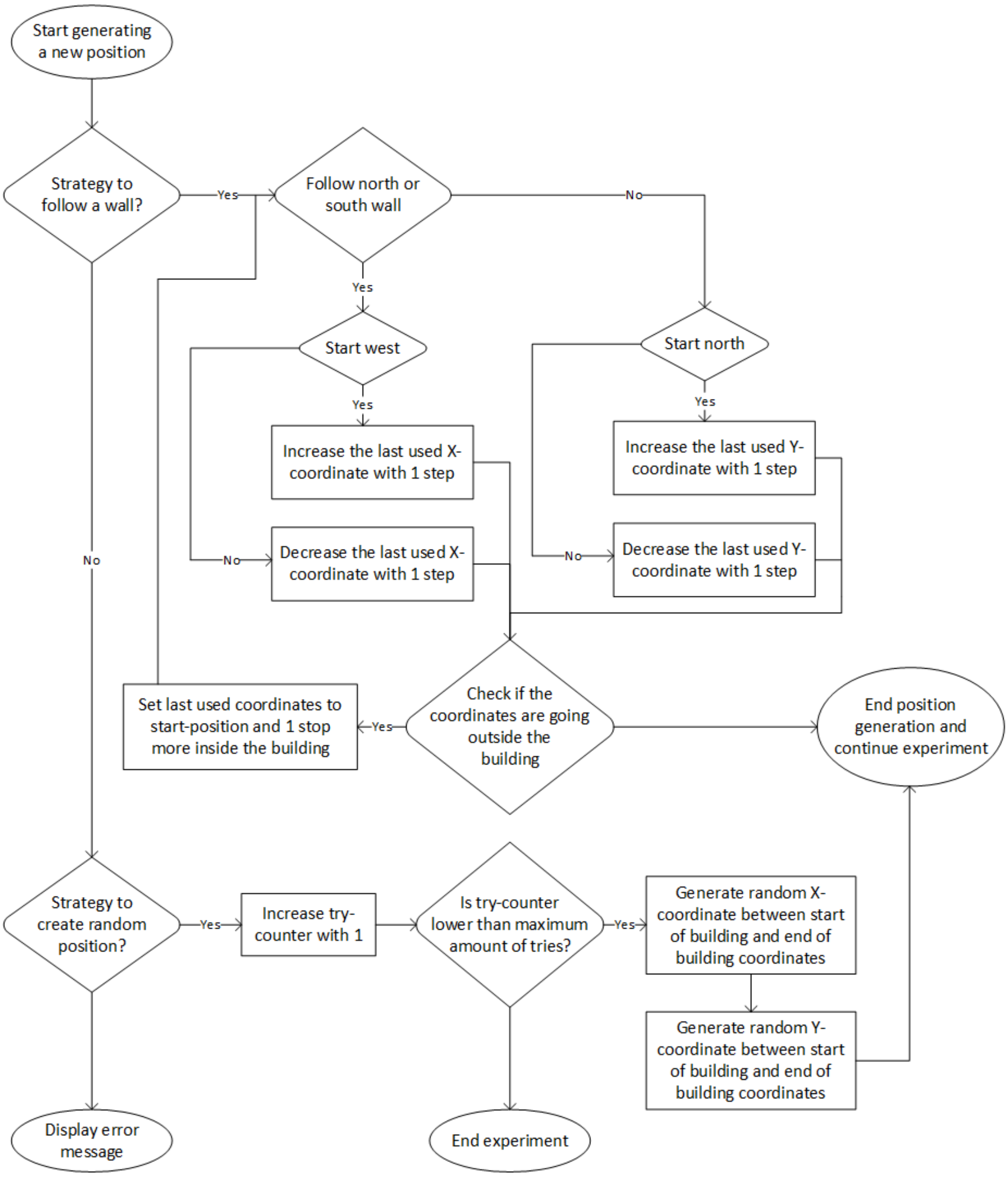
Quality Control
The quality control-frame is relatively simple. This is the final destination for the full boxes with products. It consists of one method that records all settings of the box that is brought in, furthermore it consists of a buffer with a workplace where the boxes are being dropped off. This buffer automatically pushes the boxes to a drain, removing the MU from the simulation.

Empty vessel drain
The empty vessel drain-frame works similar as the quality control-frame. This frame is the final destination for the empty vessels coming of the production machines or prefoamers. This frame also has only one method, that saves the data from that vessel. Furthermore the frame has a buffer with a workplace where the vessels will be dropped off. This buffer is connected with a drain, removing the vessels from the simulation.

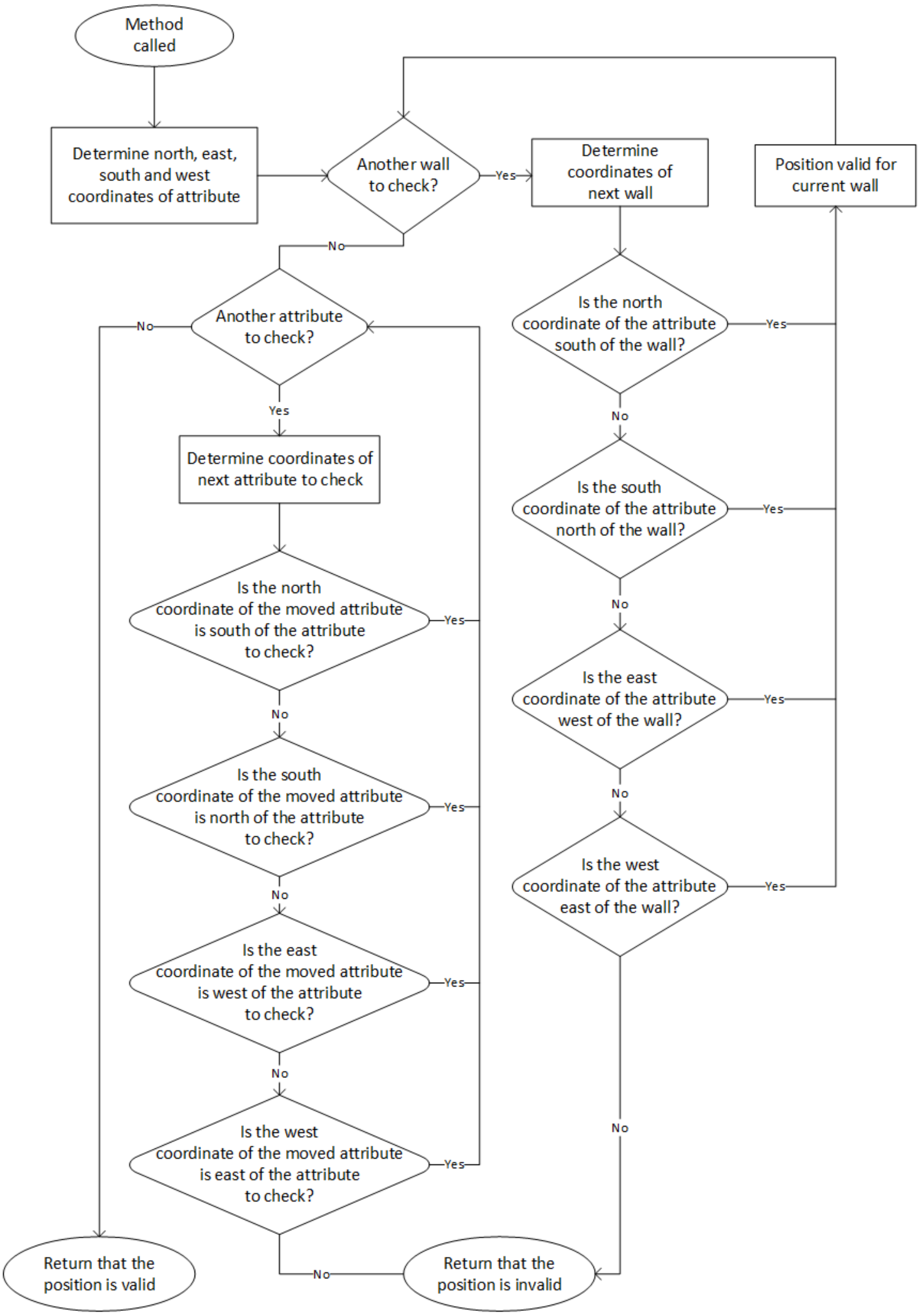
3. Flowchart



Appendix Figure 1: Flowchart of the overall simulation



Appendix Figure 2: Flowchart of the method that generates a new position

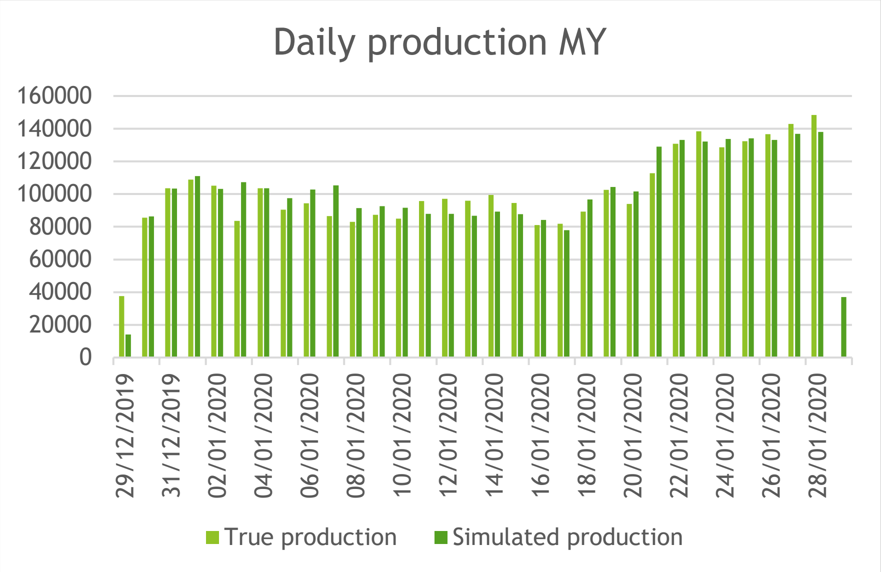


Appendix Figure 3: Flowchart of the method that checks if the given position is valid

This flowchart is activated when an attribute has been moved. This chart check whether the generated position is valid, or if the moved attribute now interference with other attributes. Attributes can be the production line, mixing machine or quality control.

4. ANOVA results

Malaysia



Appendix Figure 4: True and simulated daily production for Malaysia

Anova: Single Factor

SUMMARY

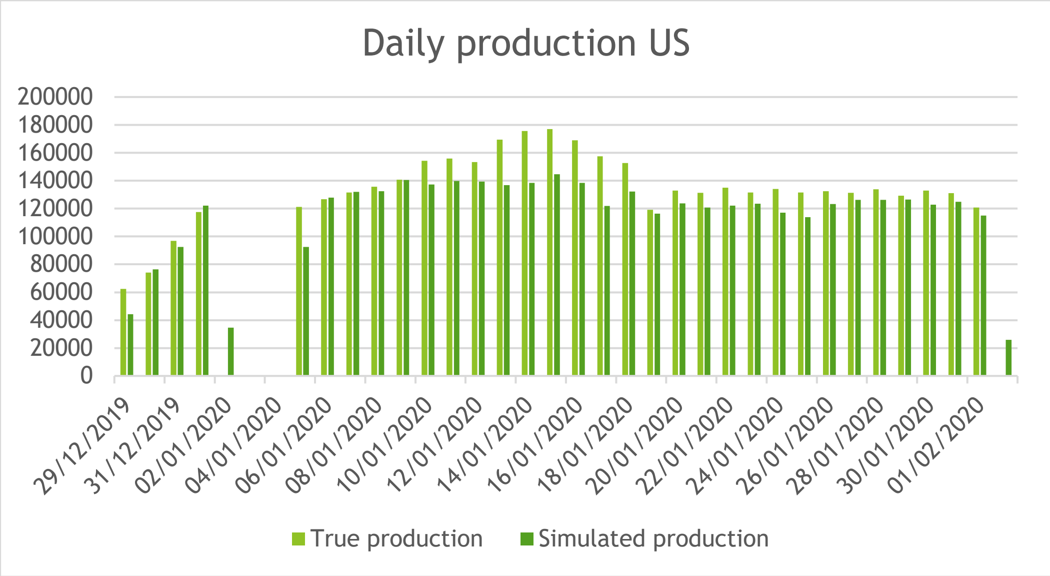
Groups	Count	Sum	Average	Variance
Column 1	30	3119301	103976,7	4,14E+08
Column 2	30	3170967	105698,9	3,56E+08

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	44489592,6	1	44489593	0,115502	0,735192579	4,006873
Within Groups	22340671659	58	3,85E+08			
Total	22385161252	59				

Appendix Figure 5: ANOVA results for Malaysia

US



Appendix Figure 6: True and simulated daily prodution for the US

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Column 1	34	4235676	124578,706	1,97E+09
Column 2	34	3881586	114164,294	1,28E+09

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	1,84E+09	1	1843819531	1,134881	0,290618	3,986269479
Within Groups	1,07E+11	66	1624681440			
Total	1,09E+11	67				

Appendix Figure 7: ANOVA results for the US

5. Comparison between simulated walk distances and straight walking distances

Production line	Machine	Number of walks	Simulation distance (m)	Straight distance (m)
1	1	23	25,50	22,5
1	2	13	31,54	23,5
1	3	22	34,25	25
1	4	16	26,74	21,25
1	5	19	26,40	21
1	6	31	23,28	20
2	1	27	19,57	17,75
2	2	24	25,41	17
2	3	14	25,82	19,5
2	4	23	23,76	17
2	5	29	17,95	15
2	6	28	15,60	13,75
3	1	20	16,60	12
3	2	29	16,96	12,5
3	3	6	22,85	13,75
3	4	20	18,90	10,75
3	5	34	17,68	9
3	6	34	14,26	7,5
4	1	28	21,34	20
4	2	26	21,40	23
4	3	28	27,27	25,75
4	4	34	28,09	27,5
4	5	27	27,51	25
4	6	30	21,44	22,5
5	1		n/a	18,25
5	2		n/a	21
5	3		n/a	23,25
5	4		n/a	25
5	5		n/a	22,5
5	6	25	21,48	19,5
6	1		n/a	17,5
6	2	2	21,47	20,5
6	3	5	30,09	23,5
6	4	34	27,93	23,75
6	5	34	22,00	20,5
6	6	12	19,13	17,5

Appendix Table 1: Simulated and straight distances for each machine in the Dutch facilities original layout

Production line	Machine	Time total simulated	Time total straight
1	1	00:15:07	00:12:56
1	2	00:10:43	00:07:38
1	3	00:19:18	00:13:45
1	4	00:11:09	00:08:30
1	5	00:13:00	00:09:59
1	6	00:18:30	00:15:30
2	1	00:13:40	00:11:59
2	2	00:15:42	00:10:12
2	3	00:09:30	00:06:49
2	4	00:14:07	00:09:46
2	5	00:13:29	00:10:52
2	6	00:11:23	00:09:37
3	1	00:08:46	00:06:00
3	2	00:12:46	00:09:04
3	3	00:03:53	00:02:04
3	4	00:09:55	00:05:23
3	5	00:15:29	00:07:39
3	6	00:12:35	00:06:22
4	1	00:15:24	00:14:00
4	2	00:14:22	00:14:57
4	3	00:19:33	00:18:02
4	4	00:24:20	00:23:22
4	5	00:19:02	00:16:52
4	6	00:16:32	00:16:52
5	1		
5	2		
5	3		
5	4		
5	5		
5	6	00:13:53	00:12:11
6	1		
6	2	00:01:32	00:01:01
6	3	00:04:13	00:02:56
6	4	00:24:12	00:20:11
6	5	00:19:10	00:17:26
6	6	00:06:12	00:05:15
Total		06:47:26	05:27:13

Appendix Table 2: Total walk times per machine for simulated and straight distance