

E.T.S. de Ingeniería Industrial, Informática
y de Telecomunicación

Definition, analysis, and optimisation of a bread production system



Grado en Ingeniería
en Tecnologías Industriales

Trabajo Fin de Grado

Jorge Valencia Antón

Directores: María Dolores Ugarte Martínez y Miguel
Iriberry Vega

Pamplona, enero 2021

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ACKNOWLEDGEMENTS

Firstly, I would like to thank my parents and my sister and express my endless gratitude to them for their continued support during these years of formation. Without their loving care and attention this journey would have been impossible.

Also, I would like to thank my teachers, especially those whose devotion for teaching their subjects was transmitted to us, the students. This implication made it easier when difficulties arose. An example of such a teacher is Lola Ugarte, who kindly accepted to co-direct this project from UPNA. I am really grateful for that.

Next, I would like to thank a person whose influence on my journey was noted from the very beginning, when he strived trying to explain to me “what is an industrial engineer” right before I decided to study this degree. He proposed me the interesting and challenging task of carrying out a project that would demand the work of a “real engineer” and based on this, he helped me figure out the topic for this project. His name is Miguel Iriberry and I want to thank him for co-directing this work, as well as the company he directs, *CONTEC Ings. Consultores SL*.

Finally, I would like to express my gratitude to *FlexSim Iberia* for allowing me to use a student license, what has improved the capabilities of the simulation software that has been used for the completion of the project (*FlexSim*).

ABSTRACT

Nowadays, companies are making good use of the increasingly developed computer technologies, such as modelling and simulation. This project focuses on the analysis and study of an existing production system, that could be present in today's industry. For that, process simulation is required. With the help of the available software, in this case, *FlexSim*, it is possible to represent the current parts and relationships within the real system by means of a virtual model that appropriately mimics the system's behaviour. Once the simulation model is ready, it will allow for the evaluation of test scenarios to the current production system. This way, the optimisation of the overall process is possible. This project intends to serve as an example of how a company might try to improve an existing process, by following the steps that are thoroughly explained over the course of this work.

KEYWORDS

FlexSim, modelling and simulation, process simulation, production system, optimisation.

RESUMEN

Hoy en día, las empresas se sirven constantemente de las tecnologías informáticas, cada vez más desarrolladas, como el modelado y la simulación. Este proyecto se centra en el análisis y estudio de un sistema de producción real, que podría estar presente en la industria actualmente. Para ello, se utiliza la simulación de procesos. Empleando el software disponible, en este caso, *Flexsim*, es posible representar las partes y relaciones presentes dentro del sistema real mediante un modelo virtual que replica de manera adecuada el comportamiento de este sistema real. Una vez que el modelo de simulación esté listo, se podrán evaluar escenarios de prueba en el sistema. De esta manera, es posible la optimización del proceso completo. Este proyecto pretende servir como ejemplo de cómo una empresa podría tratar de mejorar sus procesos, siguiendo los pasos que se explican a fondo a lo largo de este trabajo.

PALABRAS CLAVE

FlexSim, modelización y simulación, optimización, simulación de procesos, sistema de producción.

INDEX

1. INTRODUCTION	1
1.1 OBJECTIVES AND SCOPE.....	1
1.1.1 General objective	1
1.1.2 Specific objectives	1
2. BACKGROUND	2
2.1 HISTORY AND EVOLUTION OF BREAD PRODUCTION	2
2.2 MODELLING AND SIMULATION.....	3
2.2.1 Classification of models.....	4
2.2.2 Advantages and disadvantages of simulation.....	5
2.2.3 Steps in the development of a simulation study.....	6
3. PROCESS SIMULATION	9
3.1 PROCESS SIMULATION SOFTWARE	9
3.1.1 <i>FlexSim</i>	10
3.2 INDUSTRY 4.0	10
4. SIMULATION STUDY	12
4.1 BREAD PRODUCTION PROCESS DEFINITION	12
4.2 BUILDING OF THE INTRODUCTORY THEORETICAL MODEL	18
4.2.1 Verification and validation of the introductory theoretical model.....	37
4.3 BUILDING OF THE MODEL FOR THE REAL PRODUCTION SYSTEM	41
4.3.1 Deterministic model.....	42
4.3.2 Stochastic model	50
5. ANALYSIS OF TEST SCENARIOS WITH THE SIMULATION MODEL	63
5.1 TEST SCENARIO NUMBER 1	63
5.2 TEST SCENARIO NUMBER 2	67
5.3 TEST SCENARIO NUMBER 3	71
5.4 TEST SCENARIO NUMBER 4.....	74
6. CONCLUSIONS AND RECOMMENDATIONS	81
6.1 CONCLUSIONS	81
6.2 RECOMMENDATIONS.....	82
7. BIBLIOGRAPHY	84

INDEX OF FIGURES

Figure 1: Simulation Study Schematic.....	6
Figure 2: Process Diagram.....	15
Figure 3: Source.....	19
Figure 4: Arrival Style	19
Figure 5: Source’s Output.....	20
Figure 6: Processor.....	20
Figure 7: Operator	21
Figure 8: Processor’s Parameters: transport.....	22
Figure 9: Processor’s Parameters: kneading	22
Figure 10: Separator.....	23
Figure 11: Separator’s Parameters: 1st division	24
Figure 12: Queue	25
Figure 13: Queue’s Parameters: shaping of the 1st division.....	26
Figure 14: Processor’s Parameters: shaping of the 1st division.....	27
Figure 15: Separator’s Parameters: 2nd division	27
Figure 16: Queue’s Parameters: shaping of bread (1st part).....	28
Figure 17: Separator’s Parameters: shaping of bread (1st part).....	29
Figure 18: Combiner.....	29
Figure 19: Source: Tote. Shaping of bread (2nd part).....	30
Figure 20: Combiner’s Parameters: shaping of bread (2nd part)	31
Figure 21: Source: Pallet. Shaping of bread (3rd part).....	31
Figure 22: Combiner’s Parameters: shaping of bread (3rd part)	32
Figure 23: Connexions in fermentation, baking and cooling stages	33
Figure 24: Calling of task executer: Priority	34
Figure 25: Creation of a second central port	34
Figure 26: Sink.....	35
Figure 27: Separator’s Separate Mode	35
Figure 28: Separator’s Output Ports	35
Figure 29: Connexions Separators-Sinks.....	36
Figure 30: Separator’s Parameters: last step of packing.....	37
Figure 31: Arrival of 1 batch	37
Figure 32: Run Time (1 batch)	38
Figure 33: SinkBread: Properties.....	38
Figure 34: Pie Charts: Operator1 and Operator2 (1 batch)	38
Figure 35: Arrival of 11 batches	39
Figure 36: Run Time (11 batches)	39
Figure 37: Pie Chart: Operator2 (11 batches)	40
Figure 38: Pie Chart: Operator1 (11 batches)	40
Figure 39: Dispatcher	43
Figure 40: 3D layout	43
Figure 41: Time Tables	44
Figure 42: BasicFR	44
Figure 43: Operators assigned to the BasicFRs.....	45

Figure 44: Second Time Table: Members.....	45
Figure 45: Second Time Table: Functions.....	45
Figure 46: Assignment of a BasicFR.....	46
Figure 47: Night shift.....	46
Figure 48: Experimenter.....	47
Figure 49: Variable: Inter-Arrivaltime.....	47
Figure 50: Performance Measures: Bread buns produced:.....	48
Figure 51: Scenarios for Variable 1 (deterministic model).....	48
Figure 52: Experiment Run (deterministic model).....	48
Figure 53: Results: Replications Plot (deterministic model).....	49
Figure 54: Results: Raw Data (deterministic model).....	49
Figure 55: ExpertFit.....	51
Figure 56: Project Element.....	52
Figure 57: Data Analysis.....	52
Figure 58: Data Summary Table.....	53
Figure 59: Automated Fitting Results.....	53
Figure 60: Graphical Comparison Options.....	54
Figure 61: Density Histogram Plot.....	54
Figure 62: Options of Goodness of Fit Tests.....	55
Figure 63: Anderson-Darling Test.....	55
Figure 64: Simulation Representation Options.....	56
Figure 65: Simulation Software Representation.....	56
Figure 66: Picklist option.....	56
Figure 67: Using code.....	56
Figure 68: Normal distribution.....	57
Figure 69: Scenarios for Variable 1 (stochastic model).....	58
Figure 70: Experiment Run (stochastic model).....	59
Figure 71: Results: Replications Plot (stochastic model).....	59
Figure 72: Results: Raw Data (stochastic model).....	59
Figure 73: New Scenarios for Variable 1 (stochastic model).....	60
Figure 74: New Results: Replications Plot (stochastic model).....	60
Figure 75: New Results: Raw Data (stochastic model).....	61
Figure 76: Test Scenario Number 1: Time Tables.....	64
Figure 77: Test Scenario Number 1: Results: Replications Plot.....	64
Figure 78: Test Scenario Number 1: Results: Raw Data.....	65
Figure 79: Test Scenario Number 1: New Scenarios for Variable 1.....	65
Figure 80: Test Scenario Number 1: New Results: Replications Plot.....	65
Figure 81: Test Scenario Number 1: New Results: Raw Data.....	66
Figure 82: Test Scenario Number 2: Scenario for Variable 1.....	68
Figure 83: Test Scenario Number 2: Results: Replications Plot.....	68
Figure 84: Test Scenario Number 2: Results: Raw Data.....	68
Figure 85: Test Scenario Number 2: Group1: Morning shift.....	69
Figure 86: Test Scenario Number 2: Scenarios for Variables 1, 2, 3 and 4.....	69
Figure 87: Test Scenario Number 2: New Results: Replications Plot.....	70
Figure 88: Test Scenario Number 2: New Results: Raw Data.....	70

Figure 89: Test Scenario Number 3: Normal distribution	72
Figure 90: Test Scenario Number 3: Scenarios for Variable 1.....	73
Figure 91: Test Scenario Number 3: Results: Replications Plot	73
Figure 92: Test Scenario Number 3: Results: Raw Data	74
Figure 93: Experimenter: Optimizer tabs	75
Figure 94: Optimizer: Variables.....	76
Figure 95: Optimizer: Variables' Parameters	76
Figure 96: Optimizer: Objectives.....	77
Figure 97: Optimizer Run	77
Figure 98: Optimizer Results	77
Figure 99: Optimizer Results: Interarrivaltime.....	78
Figure 100: Optimizer Results: FirstDiv	78
Figure 101: Optimizer Results: SecDiv.....	79
Figure 102: Bread buns produced: Rank 4, 5 and 6	80

INDEX OF TABLES

Table 1: Available equipment.....	16
Table 2: Identification of tasks with capital letters.....	16
Table 3: Activities' duration and precedents	17
Table 4: Assignment of tasks to working stations.....	17
Table 5: Modification from the simplified case to the real available data	41
Table 6: Working schedule	43
Table 7: Modified working schedule	63

1. INTRODUCTION

1.1 OBJECTIVES AND SCOPE

In the following, it is introduced the general objective of this project as well as the specific objectives that must be dealt with to achieve it.

1.1.1 General objective

- To deliver a valid and powerful tool, in the form of a simulation model, that allows to easily understand, study, analyse, and optimise the behaviour of a real production system, a bread production system.

1.1.2 Specific objectives

- To describe the real production system that is wanted to be modelled, identifying its main characteristics, variables of interest, and all the relevant information regarding its processes.
- To analyse the data collected from the real system and to use the simulation software *FlexSim* in order to deal with this information that will eventually lead to the creation of the virtual model of the system.
- To check the validity of the simulation model created with *FlexSim*, task that can also be performed with this simulation software.
- To suggest several test scenarios as an example of any improvement proposal whose implementation in the real system could be of interest for the company.

For this project, a bread production system has been chosen, but the steps that are followed throughout this work could be applied to any real production system.

2. BACKGROUND

2.1 HISTORY AND EVOLUTION OF BREAD PRODUCTION

Since this project is focused on the study of a production system of bread, it is interesting to briefly talk about the evolution of the manufacturing process throughout the years of this precious foodstuff.

To start with, going back to the Neolithic period, our ancestors used to mix cereals with water to form a mush. This mixture was either consumed directly or left to dry in the sun. After drying, the mush became solid and turned into the most primitive form of bread [1], [2], [3] (all cited in [4]).

However, most experts agree that this primitive form of bread cannot even be considered as bread, because there is no fermentation in its production. So, they claim that it was in Egypt where bread was firstly produced, around 4000 BC, since it was there where yeast was initially included in the recipe, acting as the fermenting agent [4], [5]. The Egyptians developed and established the baking techniques and they also built the first ovens to bake bread.

It was around 800 BC when the primitive technique of grinding the grain manually with stones was improved using mills, where a rotatory movement of rounded stones was created by the effort of animals or slaves. This happened in Mesopotamia [5].

The Greek got to know about the invention of bread through their commercial relationships with Egypt. Once they adopted it into their gastronomy, the Greek started to experiment with the recipe and shape of the bread. By the third century BC, they were using shaped bread in religious ceremonies and had already created around seventy different types of bread. Some of the recipes included honey and other ingredients to sweeten the bread. This is often considered as the precursor of pastry making.

Despite how greatly spread in the Greek society the consumption of bread was, in Rome it was thought to be unreachable for regular people, only available for the most affluent spheres. However, they made significant advances in the technology of bakery: Roman engineers invented the hydraulic mill (60 BC) and further developed the kneading machines and ovens, to an extent that, nowadays, the direct heating oven is usually referred to as "Roman oven". In 30 BC, the situation with the popularity of bread had totally changed in Rome and more than 300 bakeries were opened. The ones in charge were Greek experts and, in the following years, all the bakery sector in Rome would associate and get legislated as a profession. They were so well regarded by the Emperor that they were privileged with the exemption of tax payments.

In the Middle Ages, the baking techniques kept evolving but, the crops of cereals were considerably reduced. This led to long periods of famine, since the production of bread, that had become the basis of the alimentation, was struck by the reduction of crops. Bakers, as any other profession at the time, started to organise in guilds and the production and distribution of bread was controlled by the government, something that already happened in Rome.

The agriculture progressed and better milling techniques were developed by the end of the eighteenth century, so the production of wheat increased, and the quality of flour improved. White bread was then offered to more people and the price of bread was reduced. With the invention of the vapour mill, in the nineteenth century, the baking process kept modernising as a new phase in the manufacturing was added: the airing of the dough. A new yeast was introduced, and with the advances of the mechanical kneading techniques, it was industrialised and so was the whole manufacturing process of bread [4], [5]. In that context, with the commercialisation of yeast, the first pre-fermenting doughs were developed. One of the most important was obtained by the “Poolish” method, or sponge method: original from Poland, what explains its first name, the process consists of two phases. One, a primary dough is obtained by mixing flour, water and yeast and allowing a long fermentation. Two, this dough that has become a sponge, what explains its second name, is added to the rest of ingredients, and allowed a relatively short fermentation this time [6], (cited in [7]). Another pre-fermenting dough was created in 1961, when the Chorleywood baking process was developed. This process exchanges an intensive mechanical work on the dough for the elimination of a great amount of the fermenting time, what implies a complete revolution if we compare this way of baking to the traditional one [5], [7].

So, all this evolution leads to these days, where the most important processes being carried out worldwide are the **traditional method**, the **Chorleywood method** and the “**Poolish**” **method**. This project is going to focus on a bread production system that carries out the traditional method in an industrialised way. It will be seen that some of the stages of the process include different machines that help the baker to perform his or her tasks in a shorter and more standardised time, with a smaller variability between the results produced by the same activity at different times, and with a better overall quality of each of these stages. On the other hand, it will also be seen that there are still some stages that are performed manually, without the help of any machine. It is important to understand that, in order to have a traditional baking method, some of the activities must remain being manual, otherwise, the method could not be called traditional. Yet, it can be interesting as well to identify which tasks that are still solely performed by human workers could use the help of a machine, as a way of trying to optimise the obtained results.

2.2 MODELLING AND SIMULATION

This project has the main objective of being able to understand how an existing production system is performing currently, with the characteristics of the real process, and also to predict how it would perform in the event of imposing different changes in the configuration of the system, in order to improve and optimise the behaviour of the overall process. To achieve this, the two main tools that will be used in the execution of this project, carried out with the help of a computer, are **modelling** and **simulation**.

As a matter of fact, modelling and simulation are not two separated topics: modelling is the most essential part of a simulation study. In order to perform a good simulation study, a good simulation model is required. The better the latter, the better the former [8], [9], [10], [11].

Modelling, as its name suggests, is the process of building a model, this one being a faithful representation of how a system is constructed and how it works. A good model is the one representing a system in a way it is neither too complex, what would hinder the modeller's job to handle it and understand its behaviour, nor so simple it does not include the most important features of the actual system. That is, a trade-off is intended in between realism and simplicity.

A simulation study of a system of interest is performed by letting the system's model operate. This is what makes simulation so valuable: based on the behaviour of the model, it can be inferred how the real system would behave under similar conditions to the ones imposed on the model. This allows the person in charge of the simulation study to experiment and modify the configurations of the model, and to find out how the performance of the system would vary depending on those changes being applied to the simulation model. Without simulation, it would be necessary to directly apply the changes to the real system, what, in most cases, will be too expensive and not practical at all. In today's industry this is not even negotiable: you do not implement a change in a real system unless you have completely proved, by means of simulation studies, that this modification implies a clear benefit on the performance of the original system. Also, simulation does not only apply when some modifications are intended in an existing system, it is the starting point for the creation of a whole new real one. A company that manufactures engines, for instance, will always test a model of the actual product before building it. A company cannot take the risk of expending lots of money in building a product that, eventually, might not work as expected. In practice, simulation is the solution.

2.2.1 Classification of models

Computing has experienced a great development over the last few years, especially in the areas of decision making and process and product design. Of course, simulation has been indispensable in this progress [11]. Nowadays, it is quite accessible to model almost any real system and to make decisions based on the results provided by the great deal of simulation software currently available. These models might be either **physical** or **mathematical** models [8], [9], [11].

- A physical model is the one representing a system by means of a scaled object of the original one that is put under equivalent physical conditions to the ones present in the actual system being represented. For instance, the performance of a plane when experiencing turbulence or a building located in a very windy emplacement will be simulated by means of a Wind Tunnel test.
- A mathematical model is the one that is more often used in a simulation study. It works by the action of mathematical and statistical equations that determine the logical and probabilistic relationships which define the complete behaviour of the system being modelled. There is a particular group of models inside the mathematical ones whose simulation is called **Discrete Event Simulation (DES)**, where the system presents instantaneous changes in response to certain discrete events [8]. DES will be further explained along the course of this project.

Inside mathematical models, we can distinguish two great groups, depending on the type of mathematical equations that define their behaviour.

- **Continuous models**, where it is possible to evaluate the performance of the system in a continuous time span. This is fulfilled using differential equations to characterise the key relationships between elements in the system. A model containing the flow of a fluid element is an example of a continuous model.
- **Discrete models**, where the evaluation is carried out at specific moments of the total time span. That is, evaluating equations at specific points.

Another classification criterion has to do with the influence of the evolution of time in the simulation of a model.

- **Static models**, where time does not affect the simulation. The simulation of these models is generally performed using the Monte Carlo method. The origin of its name is the roulette, the most popular casino game, where 37 empty pockets await the arrival of a ball introduced by the croupier. There are only 37 possible solutions when “simulating” this “model”, and this is always the case, no matter how much time passes.
- **Dynamic models**, where the system being represented changes as time changes.

Also, depending on the nature of the variables that are being considered by the model,

- **Deterministic models**, where all the relationships and changes between variables are fixed and known.
- **Probabilistic or stochastic models**, where the interaction between the different parts of the model depends on statistical functions. Randomness is key in this type of models.

This project is focused on the study of a complete production system, what means it is necessary to model and simulate the whole process, from the arrival of the raw materials to the departure of the finished products and taking into account the interaction between the different machines and workers within the system. Therefore, **process simulation** is required.

Process simulation will be discussed later, when it is introduced the main software that has been employed to develop this project: *FlexSim*. For the time being, it is enough to understand that the best way to simulate a production process like the one concerning this project is DES, whose models will be mathematical, discrete, and dynamic, with both deterministic and stochastic variables [11].

2.2.2 Advantages and disadvantages of simulation

In general, the use of simulation provides a great deal of **benefits** [8], [10], [11].

- a) Allows the modeller to get to know the process better and to understand how it is modified by the impact of changes in the model, without implementing them in the real system.
- b) Provides different scenarios, from which the person in charge will determine which one is the most advantageous.
- c) It is easy to identify bottlenecks or any other problem that might arise.
- d) Time can be compressed, allowing to observe any behaviour during a long period of time in a much shorter time span. Also, time can be expanded, allowing the watcher to analyse in detail an interesting behaviour over a longer lapse of time.
- e) Great and user-friendly software is available and keeps developing. It is becoming easier to use and more powerful, being able to solve increasingly difficult problems.

However, in the process of developing a simulation study some **pitfalls** might be encountered as well.

- a) Sometimes, an analytic solution is enough. Simulation will provide a less accurate solution in these cases.
- b) The model is invalid, it does not correctly represent the system it is simulating. The invalidity of a model can be due to a lot of reasons, for instance, being too simple or too complex.
- c) In some advanced studies, the required time to build a model or qualification to handle the simulation software might be too demanding. Not meeting the requirements guarantees failure.
- d) It might not be clear what variables to monitor, how to implement the different probability distributions or which assumptions are relevant to be made at each stage of the modelling process.

2.2.3 Steps in the development of a simulation study

Firstly, it is important to understand the emplacement of a simulation study, in relationship with the real system it is modelling and intends to simulate. This is schematically represented by the following figure.

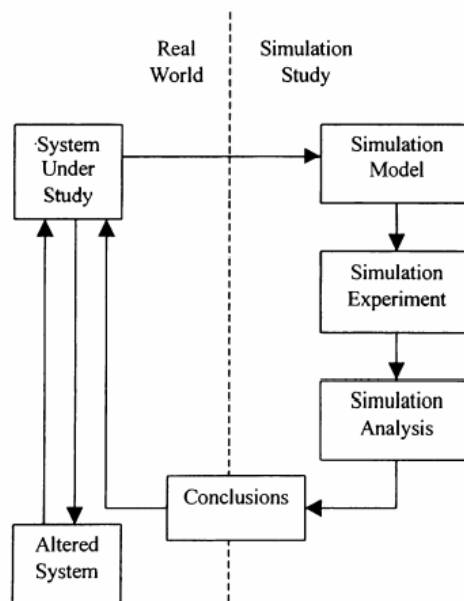


Figure 1 [8, fig. 1]

The *System Under Study* is modelled and simulated. Once the *Conclusions* are extracted from the *Simulation Analysis*, the studied system gets modified by whatever the analyst considers that will enhance its performance, with the result of this *Altered System* becoming the new *System Under Study*. This process can go on and on [8].

It is now interesting to take a deeper look at the different steps that are necessary to be followed to perform a successful simulation study, since these are the same **steps** that the simulation study proposed in this project will need to go through [8], [9], [11], [12].

- 1) **Definition of the *System Under Study*.** In this first step, it is necessary to get to know about the system to be modelled. To identify and bound the parts to be studied: the problem, so to speak. The decision variables and performance measures must be defined, and it must be possible to establish an initial conceptual model, representing the process flow and interactions between parts.
- 2) **Collection and analysis of real system data.** Here, real system's data must be collected and interpreted. It is determined which variables are deterministic and which are stochastic. The better the analysis, the better the representation of the system.
- 3) **Modelling.** Basically, by means of simulation software, to turn the initial conceptual model into a software acceptable form, and to include the characteristics considered in step 1) and all the data collected in 2). The model is now complete.
- 4) **Verification of the model.** Once the model is finished, it must undergo two checks. The first is verification, where the model is simply run to observe if all its parameters work as expected.
- 5) **Validation of the model.** The second is validation. The performance of the model under known conditions is compared to the real system's performance. If the results provided by the model are reasonably similar to the ones obtained in the real system, the model is valid for performing a simulation. In case the simulation model has been developed as a prior step to the building of a new real system, results in the model cannot be compared to results in the system, since it does not even exist. So, the validation will need to be based on the comparison of the simulation results to the expected results of the actual system in a series of experiments under certain initial conditions.
- 6) **Definition of test scenarios.** Now that a valid model is available, different scenarios must be defined to simulate the performance of the model in each of them. Most simulation software nowadays includes the option of applying an optimisation tool to the model being simulated, whose use would be helping the simulation process to produce increasingly better test scenarios.
- 7) **Interpretation and analysis of the results.** Based on the data obtained in the simulation (and optimisation, in case the tool was used together with the simulation), scenarios can be compared, and the best choice can be found. There might be more than one scenario with interesting results, in which case it is the task of the analyst to pick one over another.

- 8) Documentation of the model, conclusions, and recommendations.** It is important to keep track of the whole process, in terms of documentation. But documenting precisely this last step is essential since it can be the starting point of a future study. Finally, conclusions and recommendations are a key point in the whole simulation study. As a matter of fact, in this project, this last part is what really shows what the use of performing a simulation study is: it allows the author to deliver his or her own contribution to the whole work, by getting to different conclusions based on the interpretation of the results, and by providing recommendations, in accordance with the obtained conclusions.

3. PROCESS SIMULATION

3.1 PROCESS SIMULATION SOFTWARE

As it was mentioned earlier, the main idea of this project is the study, analysis, and simulation of a complete production system, what implies **process simulation** is necessary.

All the information that has been provided about simulation is also valid for process simulation, but it is interesting to note the difference between these two, in order to understand why the latter is required. Simulation, as it has already been discussed [8], [9], [10], [11], has to do with the performance of the simulation model, which represents any real or theoretical system that belongs to the real world. On the other hand, process simulation is, basically, the application of a simulation study to a system that includes several parts, whose individual behaviours will alter the behaviour of their fellow parts. Therefore, the importance of process simulation is its ability to accurately analyse the performance of a complex system, by carefully considering the interrelation between all its parts.

It has also been stated that the approach for the performance of the intended simulation study is **Discrete Event Simulation (DES)**. However, DES is not the only way to confront this type of problems: **System Dynamics (SD)** is a different approach that is normally used, together with DES, to build simulation models that represent systems where processes are involved [13]. Both DES and SD started developing, independently from each other, in the late 1950s. Nevertheless, the interest in comparing these two approaches, or even combining them, is quite recent. To start with, provided that the model is well executed by the modeller, both DES and SD should provide equivalent solutions to a proposed problem. However, it will be more appropriate to use one or the other, depending on the nature of the system to be modelled and the specific interests of the modeller [14], [15]. Then, why is it chosen to use DES in this work and not to use SD instead?

The main difference between DES and SD is that the first represents changes at discrete points of time (by means of **discrete models**), whereas in the second, state changes occur in a continuous fashion (by means of **continuous models**). DES allows to represent entities, that is, the entries to the system (in the form of objects, people...), individually and keep track of them throughout the process. SD, on the other hand, represents entities as a continuous quantity of stock. DES models are mainly **stochastic**; SD ones are generally **deterministic** [14], [15]. Based on the aforementioned characteristics of DES and SD it is easy to figure out that the most appropriate approach to be used in this project is Discrete Event Simulation.

Once the main idea of process simulation and the reasons why DES is the choice for performing the simulation study have been discussed, it is time to talk about the available software to carry out process simulation by means of DES. Some general ideas concerning this software will be introduced and, a deeper look will be taken into the specific software that has been employed: *FlexSim*.

Nowadays, most companies around the world have implemented to their research and development departments different software to deal with the activities that might use

simulation as a means of improvement. DES software is widely used by companies who work around manufacturing processes, what is not surprising considering that this kind of systems normally consists of the succession of several discrete events, in order to turn an initial raw material into a finished product. Not only in manufacturing, DES has become really popular in the health care sector [16], since it allows the decision-makers to modify the existing systems by taking into account all the different variables that might be hindering the process (for instance, rate of arrivals or rate of service). Basically, all the available DES software can perform useful modelling of real systems, so, in most of the cases, it will be up to the modeller to choose one software or another to carry out a simulation study. However, despite having similar main characteristics, each software has its individual and differentiating features, which eventually will convince the user and help him or her to choose between a great deal of possible options. For example *AnyLogic*, *Arena*, *Emulate3D*, *FlexSim*, *ProModel*, or *SimEvents*.

3.1.1 *FlexSim*

The chosen DES software to develop this project has been *FlexSim*. But why has it been chosen? This simulation tool is really powerful and easy to use, with the great advantage of allowing the modeller to work on a 3D environment from the very beginning. The drag-and-drop interface, that allows to take any item from the libraries and include it directly into the 3D setup, helps the modeller to accurately replicate the real system without jeopardising a successful analysis of the model. Also, as it was stated previously, most simulation software packages include an optimisation tool that helps them choose the best scenarios. In that regard, *FlexSim* is completely integrated with a powerful optimisation package, named *OptQuest* [17], [18], [19]. Therefore, based on the requirements for the completion of this work, *FlexSim* seems to be a sensible choice.

The website *Descreye Solutions*, which provides a simulation software tool itself (*OPS (Online Process Simulator)*), presents a ranking of the best available DES software nowadays [20]. *FlexSim* is crowned as *WINNER* in each of the five categories being considered: capability, purpose, ease of use, popularity, and development.

As a little introduction, and without getting into much detail, a model in *FlexSim* is basically a flow of entities (flow items) through a system where they might encounter waiting places (queues) before entering different processes (processors) and where they are moved into the different resources by means of transportation systems [17], [18], [19]. When the definition of the model is presented, later on in this project, it will be explained more specifically how the different tools that *FlexSim* provides are to be used in order to create a good simulation model.

3.2 INDUSTRY 4.0

One of the most interesting attributes of *FlexSim* is that it is leading the way in a lot of the most important areas of the rising and increasingly popular Industry 4.0 [17]. This term has been coined to refer to the fourth industrial revolution that is currently taking place in this sector [21], [22]. The first occurred thanks to the steam engine, the second with the development of the assembly line, and the third, also known as the digital revolution, thanks to the computerisation that started in the 1950s. Analog technologies became digital and, with the help of electronics,

production started to be automated [23]. In this context, the Industry 4.0, often referred to as Connected Industry, has just emerged and plans to take the digital era started with the third industrial revolution into an interconnected future.

The main contribution of Industry 4.0 technologies is that they allow to deeply interrelate the physical and digital worlds, what allows to turn any existing system into a smart and fully automated one. This duality of worlds is represented by means of **cyber-physic systems**, where any physical thing inside a system is monitored using automated computer-based algorithms [21]. What is more important, these systems can communicate between them, without the intervention of a human operator. This feature is a central pillar in the Connected Industry, and its named **Internet of Things (IoT)**. Now, some of the Industry 4.0 technologies that *FlexSim* software includes will be briefly discussed [17], [21].

- **Simulation and Digital Twin:** simulation has already been explained in detail. A digital twin is a further developed simulation model. It must be able to communicate back to the real system and allow this one to change automatically in order to meet the requirements being delivered by the model. That is, IoT interaction between the simulation model and the real system.
- **PLC Emulation:** a Programmable Logic Controller is a device which is used to control certain stages of a manufacturing process, like assembly lines or robots [24]. *FlexSim's* tool allows to emulate the PLC functionality and validate or optimise the controls inside a production process.
- **Cloud Computing:** cloud computing is becoming increasingly necessary. *FlexSim* permits to run simulation models through its web server technology and to run lots of replications using distributed CPUs.
- **Virtual and Augmented Reality (VR and AR):** extended reality is truly relevant in applications like technician training, maintenance, or safety.
- **Big Data + Data Analytics (AI):** *FlexSim's* solutions provide some valuable data that can help foresee the future of a system, or even understand how it would change if it were put under different circumstances. Artificial Intelligence is a key tool in the data analytics strategy, what helps the system become more competitive, something that is really important if we extrapolate any system into the real industrial world.

Apart from these technologies that can be implemented with *FlexSim*, Industry 4.0 includes others like **additive manufacturing** (3D printing), **collaborative robotics** or **cybersecurity** [21].

4. SIMULATION STUDY

4.1 BREAD PRODUCTION PROCESS DEFINITION

Now, it is time to define the production system that has been chosen to be the initial stage of the intended study. This definition will be based on the real company that has been analysed in [25].

The starting point of the process is the production order that is carried out by the production manager, who needs to make sure that the amount of available raw materials for the bakers to produce bread is enough to meet the demands of the different establishments to which the finished product will be supplied. Once the amount to be produced is known, with the required raw materials being stored in the production area, the production process can begin.

The bread production process that is being studied, which could represent any real process that produces bread according to the traditional method and with the help of machinery in certain stages, can be divided into the following parts: weighing, kneading, dividing of the dough, shaping, painting, fermentation, baking, cooling, and packing.

- **Weighing**

The materials are carried from the warehouse to the scale, where the worker will weigh each of the ingredients so that the dough obtained from the mixing of these is prepared according to the recipe, depending on the specific type of bread. Thus, the quality can be kept constant between the different batches of the same product.

- **Kneading**

Once the materials' weighing is completed, they are introduced in the kneading machine. Here, they are firstly mixed, and afterwards, once a uniform dough has been obtained, it is kneaded. The operator is in charge of transporting the ingredients to the machine and then, after a 15-minute average kneading that is performed without the operator's help, he or she must collect the dough from the machine and head towards the following task.

- **Dividing of the dough**

This division is performed in two steps: firstly, a worker uses a divider machine to turn the dough into big, somewhat irregular balls. These balls are shaped before they can enter the second step of the division, where a balling machine turns the big balls of dough into smaller balls, which can be much easily handled. Once these last balls are ready, they can be delivered to the master baker.

- **Shaping**

As mentioned above, this task is performed by a master baker. It is in this step where it becomes possible to emulate the bakery techniques from past generations, what allows the process to still be called “traditional”. Nowadays, there exist plenty of options to perform this shaping with the partial or total implication of machinery. Of course, the levels of bread production in the companies that employ these tools is much greater. However, it must be given great importance to the quality of the product and to the customers’ preferences, what can make a company opt for the traditional by hand shaping, and not for the more industrialised one.

Shaping is key in determining what type of bread is being produced. Moreover, there are cases in which the types of bread are told apart from each other, basically, by their shape since their ingredients are practically identical. So, it seems reasonable that a master baker must have the required capabilities to produce any shape being demanded, with the appropriate size as well. Otherwise, the client might not be happy with the product being offered.

The shaped bread is put on trays which are introduced in trolleys. Whenever a trolley is ready, it can be taken to the fermentation chamber.

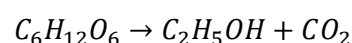
- **Painting**

Depending on the variety of bread, it can require a painting layer that is performed manually by the master baker. This layer is the one giving the specific type of bread its characteristic final colour, the one that can be observed after the baking process. This painting is performed directly on the bread already located on trays, before introducing them to the trolleys.

- **Fermentation**

As it has been stated previously, fermentation is what allows bread to be called as such [4], [5]. A fermentation chamber must meet all the required conditions of temperature and moisture to allow the yeast present within the dough to react as expected, what increases the volume of bread and prepares it to get baked in the following step.

The effect of the fermentation of yeast can be briefly summarised as follows: yeasts behave somewhat like human beings, transforming carbohydrates into energy that is later used in other processes [26]. So, starting from glucose ($C_6H_{12}O_6$), which is a carbohydrate, a series of chemical reactions eventually yields ethanol (C_2H_5OH) and carbon dioxide (CO_2). The overall reaction would be something like this.



In the case of fermentation for bread production it is the realising of carbon dioxide what interests. This gas enables the leavening of the dough, what provides the desired volume to the bread [27], [28], [29] (all cited in [30]).

On the other hand, there can be cases where the fermentation product that is of interest is the ethanol, commonly referred to simply as alcohol. This is the case, for instance, of beer production.

Therefore, it could be said that bread and beer are somehow similar in terms of both their production processes having yeast fermentation as their pivotal step. Yet, it is easy to see that once the fermentation starts, the two paths that are followed by each of them start diverging, ending up with two very different products [30].

In the system being studied, the average time that a trolley must remain inside the fermentation chamber, so that the yeast properly develops, is 30 minutes.

- **Baking**

The trolleys that leave the fermentation chamber are ready to be baked. This process consists of introducing them in an oven for an average time of 15 minutes, depending on the variety of bread, at a temperature that cooks the bread according to the recipe. When a trolley leaves the oven, provided time and temperature have been programmed appropriately, bread production is complete.

- **Cooling**

The high temperatures that have been reached inside the oven to bake the bread prevent the workers from being able to directly proceed with the packing of bread into boxes. An intermediate step is necessary, where with the help of a fan, the bread is cooled down for around 15 minutes. After the cooling, workers can start packing.

- **Packing**

The operators will need to pack the bread following the instructions of the production manager, so that all the deliveries are carried out correctly and as quickly as possible. It is important to understand that timeliness is very valuable for customers, so a smart management of this last activity can be a guarantee of success.

It is also true that a delay in the delivery of an order can be caused by something other than the packing step. In that case, it will be necessary to identify the source of the problem as soon as possible, and act over it in order to fix it.

Once it has been explained the whole process, it can be interesting to represent a process diagram so that the flow of the process is clearly understood [25].

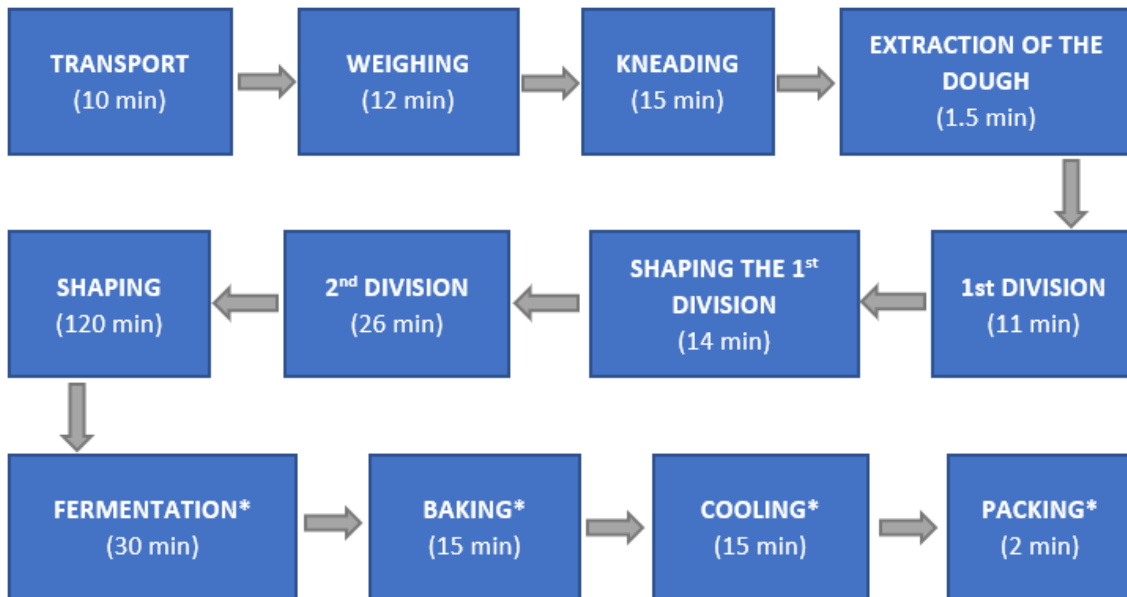


Figure 2

Some extra activities have been included in the process diagram above. Basically, the weighing process intrinsically included the transport of raw materials from the warehouse to the scale. This division simply defines the separation of one task from the other. The extraction of the dough from the kneading machine was discussed as the last step of the kneading process and its dividing was performed in two steps which, after introducing the shaping of the first division of the dough as a step itself, eventually have become three. On the other hand, the painting activity has been removed since it has been chosen to produce a variety of bread that does not require this step.

Also, in the activities where a * has been included after the name, the process time indicated below refers to a single trolley. So, it will be necessary to know how many trolleys are required to fulfil a complete batch and how much room is available inside the fermentation chamber, as well as the baking and cooling capacity of the ovens and fans, respectively. It will be explained in detail later, but for now, it is enough to assume that **4 trolleys** are required to fulfil a complete batch.

For the complete process, the available machinery will be listed next. The use of the machines will either help the worker to complete a task, or allow the completion of a step where a worker is not required.

Equipment	Amount	Description
Weighing scale	1	Allows the worker to weigh all the components, according to the recipe
Kneading machine	2	Mixes the ingredients and kneads the dough, without the help of a worker
Divider machine	1	Helps the worker to start the dividing of the dough
Balling machine	1	Helps the worker to finish the dividing of the dough
Fermentation chamber	1	There is room for 4 trolleys, does not require a worker
Oven	2	Room for 1 trolley each, does not require a worker
Fan	1	Capability of cooling 2 trolleys at the same time, no worker

Table 1 [25, tab. 2.2]

Next, it would be of great interest to determine how much time it is required to produce the first batch of products. This can be easily obtained by adding the duration of all the stages that have been introduced in the process diagram. There is still one consideration to bear in mind for this computation to make sense: the shaping stage takes 120 minutes to be fully completed, that is, every 30 minutes, 1 trolley that is full of shaped bread is ready to go on to the baking process. Therefore, it makes sense to add to the total 120 minutes of shaping the duration of the fermentation, baking, cooling, and packing of **just 1 trolley**, since by the time the fourth trolley leaves the shaping stage, the other three trolleys are either already packed or undergoing the last processes before packing. The computation would be as follows.

$$\begin{aligned}
 & \textit{Time for the completion of the first batch (minutes)} \\
 & = 10 + 12 + 15 + 1.5 + 11 + 14 + 26 + 120 + 30 + 15 + 15 + 2 \\
 & = \mathbf{271.5 \textit{ minutes}}
 \end{aligned}$$

Once the first batch is complete, the theory tells that the time to complete the following batches, also known as the **cycle time**, is precisely the time it takes to overcome the activity representing the **bottleneck** of the process. The bottleneck of a production process is the activity that can be performed with the smallest frequency or, equivalently, the one with the longest duration. In the case of the process being studied, the bottleneck would be the shaping stage and, therefore, the cycle time will be equal to **120 minutes**. Knowing the cycle time that could be achieved to produce batches, it is important to find out how many workers would be necessary to be able to keep up with this rhythm of production. To do that, it must be solved an ALBP, or Assembly Line Balancing Problem [31]. By means of this ALBP, it will be possible to assign tasks to different working stations, in order to meet the requirement of a cycle time of 120 minutes. For the sake of simplicity, capital letters will be assigned to the tasks.

A	B	C	D	E	F
Transport	Weighing	Kneading	Extraction	1 st division	Shaping the 1 st division

G	H	I	J	K	L
2nd division	Shaping	Fermentation	Baking	Cooling	Packing

Table 2

Activity	A	B	C	D	E	F	G	H	I (4)	J (4)	K (4)	L (4)
Time (min)	10	12	15	1.5	11	14	26	120	30	15	15	2
Precedents	-	A	B	C	D	E	F	G	H	I	J	K

Table 3

Based on the table above, the balancing can be carried out following three basic rules to assign the tasks to a working station [31].

- 1) The task that is trying to be assigned has not been assigned yet.
- 2) Its precedent tasks have already been assigned.
- 3) The assignment of a task to a station does not imply that the working time of that station becomes greater than the cycle time (that is, the time of the task is not bigger than the available idle time inside the station).

The assignation is performed according to the following table.

Working station	Activity	Working time (min)	Idle time (min)
1	A, B, C, D, E, F, G	89.5	30.5
2	H	120	0
3	I (4 times)	120	0
4	J, K (4 times each)	120	0
5	L (4 times)	8	112

Table 4

As it can be deduced by analysing the tasks being performed inside the working stations, number 3 and 4 are fully composed of equipment. Therefore, it is seen that **3 operators**, the ones corresponding to number 1, 2 and 5, are needed so that the level of production of 1 batch every 2 hours is possible, following the current configuration of the system. It is true that, according to the rules that were imposed for the assignment of tasks, 3 workers are needed; however, it can be seen that 2 out of the 3 operators have a lot of idle time still available inside their stations. Therefore, it seems reasonable that the worker in station 1 could be asked to perform the task L, currently assigned to working station 5. The new configuration would lead to the disappearance of this last working station and to the reduction of the idle time in station 1, from 30.5 minutes to 22.5 minutes.

Furthermore, task C, which has been assigned to working station 1, is not performed by the worker since it represents the kneading of the dough, that is completed by the only action of the kneading machine. This means that, during the 15 minutes that the machine is kneading, the operator would be available to do other activities. So, the idle time of this worker eventually turns out to be equal to 22.5+15 minutes, what yields **37.5 minutes**. That is, by readjusting the previous table, it can be said that the cycle time of the process will be met by means of **2 workers**, one devoted strictly to shaping the bread, and another in charge of performing all the remaining tasks that require the use of a human worker, what will still allow the latter a total idle time of 37.5 minutes per cycle.

4.2 BUILDING OF THE INTRODUCTORY THEORETICAL MODEL

After the resolution of a first theoretical approach to the system being studied, by means of the discussed ALBP (Assembly Line Balancing Problem), it has been decided to build a model of the system according to this theoretical solution that has been obtained for the assignation of tasks to operators in different working stations. This model, at the same time, will work as an introductory model to what will eventually become the model of the real production system. As it has already been discussed, the satisfactory building of this final model of the system will be the key to a successful analysis of the different alternatives or improvements that will be proposed as a result of the simulation study itself.

To start building the model, it will be necessary to explain how the tool that is going to be used, which is *FlexSim*, must be handled. This software will provide a series of items in the form of 3D objects which will need to be interrelated to each other and which, by means of a smart use of the different factors that characterise all these items, will be able to properly mimic reality. There are three basic categories of items in *FlexSim* [17], [18], [19].

- **Flow Items:** they represent the elements flowing through the simulation model, from the beginning to the end. That is, in a real system like the one being represented, the raw materials before entering the kneading stage, the dough, the divisions of the dough and the shaped and baked bread will be represented by flow items.

- **Fixed Resources:** they represent the elements that would be static within a production process. Static means that the element does not move along the process, that is, it is located at a fixed position. Fixed resources represent the different stages through which the flow items must pass to complete the whole process. These would include the *queues*, where the flow items wait to enter a machine or *processor* (it will be seen that some machines are able to combine or separate elements, apart from processing them, what will be modelled with the help of *combiners* or *separators*, respectively), and the entries and exits of the flow items, that will be represented by means of *sources* and *sinks*. Also, if a machine can perform several activities in a sequential mode, it will be represented by a *multiprocessor*.

- **Task Executors:** these items can move along the system and interact with both flow items and fixed resources. As their name suggests, they can perform tasks or task sequences they are assigned, such as carrying flow items from one fixed resource to another, or operating machinery. They are normally “shared” between different fixed resources and they represent the workers and transportation machinery of flow items inside a system.

There are other types of objects that could improve the simulation model, but they will not be discussed and will only be explained in case they are required in a further step of the building of the model. Visual objects, conveyors, warehousing, AGV, or fluid objects are some examples.

Now that it has been introduced what items will be employed to build the model in *FlexSim*, it is time to start explaining what the considerations must be so that a successful model is attained.

- 1) To start with, it will be dragged-and-dropped a **source** from the *FlexSim* library into the 3D setup. As it was mentioned previously, this is one of the great advantages of *FlexSim*, since the modeller will be able to easily understand how the process goes while he or she is building the model in an intuitive and precise way. This drag-and-drop action will be repeated any time the modeller wants to introduce another item from the library into the model.

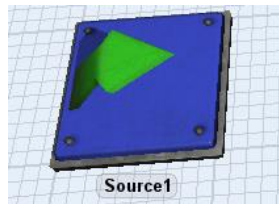


Figure 3

The source parameters allow the modeller to choose the rate at which flow items are created (*Arrival Style*), as well as the *FlowItem Class*, which has to do with the geometry of the flow item being created. By default, a source will create a brown box. It will be possible to modify the 3D geometries of all items inside a *FlexSim* model but, for the time being, the default geometries will be used.

The *Arrival Style* can be expressed by means of an *Inter-Arrival Time*, that might be fixed or based on a statistical distribution, or following a certain *Arrival Schedule* or *Arrival Sequence*, for what it would be necessary to express the desired arrival times on a table (*Edit Table*). The *Arrival Style* for this case will be determined later, after this whole introductory model is complete.

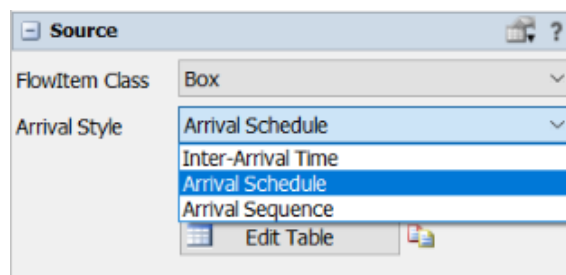


Figure 4

Based on the characteristics of the system under study, it is known that the first stage of the production process is the transportation of the raw materials from the warehouse to the weighing scale. Since the available data for the time for this process gives the total amount of time that takes a worker to perform the whole activity, it has been decided to consider 1 flow item as the representation of the complete batch of raw materials. This way, any time the source creates a flow item, it will be allowing the entry to the system of the total amount of raw materials needed to produce 1 batch of bread.

Finally, to allow the source to be joined to the following objects in the model, it will be necessary to create a “path” for the flow item to follow. This path is represented by the creation of **output-input port** connexions. By using the *Connect Objects* command or, even easier, by pressing the A key while clicking on the first of the two objects to be connected and keeping it pressed until another click is done at the second, a connexion is achieved by having created an output port at the first object and an input port at the second one. The source’s output port will be sending a flow item (*Send to Port*) to the first available input port (*First available*), option that is selected by default.



Figure 5

- 2) Knowing that 1 flow item will be representing all the ingredients and that the given data for the 4 initial stages, which are transport, weighing, kneading and extraction of the dough, correspond to the total time of these 4 activities, it is quite easy to continue with the modelling of this steps of the process, because they only require the use of **4 processors**.

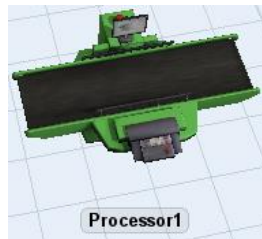


Figure 6

Processors include several parameters, like *Max Content*, what express the maximum number of flow items that can be processed at the same time, and *Setup Time* and *Process Time*. The *Setup Time* is used whenever a machine requires a certain time for preparation or configuration before the process itself, and *Process Time* strictly refers to the time the processor requires to fulfil a task. There is an important feature about these times inside a processor: they might or might not require the use of an *Operator*. In case they require one, the operator will have to work together with the machine in the completion of the tasks being represented by the *Setup* and *Process Time*. In case the operator is not needed, the worker will not have to perform the activities with the machine and could use this time to perform other tasks that he or she might have been assigned.

So, for the case of the 4 processors that are going to model the 4 activities, it has been described how they are performed and, therefore, stated which ones require the presence of a worker. The transport, weighing and extraction of the dough from the kneading machine use an operator to perform the tasks, whereas the kneading process is performed automatically, without the help of a worker.



Figure 7

In order to assign a worker to the activities that require one, it will be necessary to “call” the operator from the processor, for what a **central port** connexion must have been previously created. Central ports can be built in a similar fashion to how the output-input ports are created: using the *Connect Center Ports* command instead of the *Connect Objects* one or using the S key instead of the A key between the two objects that are wanted to be joined by a central port connection. One of these two objects will be a fixed resource and the other one will be a task executor. By construction, the only one having a central port is the fixed resource, therefore it does not matter in what order the connexion is created, it will mean the same anyway. For the case of output-input port connexions, it was explained that the order was key in the correct definition of the model since these connexions are the ones defining the route to be followed by the flow items throughout the system.

As nothing has been mentioned regarding the setup of any of the stages, all their duration will be constituted by the *Process Time*.

Then, the parameters to properly model each of the 4 activities by its corresponding processor will be determined by a table similar to the following, where clicking the white box beside *Use Operator(s)* allows the processor to call for an operator, in this case, the one connected to the central port number 1 (*centerObjects[1]*). The table expresses the parameters of the processor for the **transport** stage, where 600 second represent the 10 minutes provided by the data. Seconds will normally be used as the time unit in *FlexSim*, so the parameters for the **weighing** and **extraction of the dough** will just use the same table beneath, but with a *Process Time* of 720 and 90 seconds, respectively. On the other hand, *Priority* and *Preemption* have to do with the order in which the task executor performs the tasks they have been assigned. When the time comes, it will be discussed how to employ these parameters in order to ensure the correct behaviour of the model.

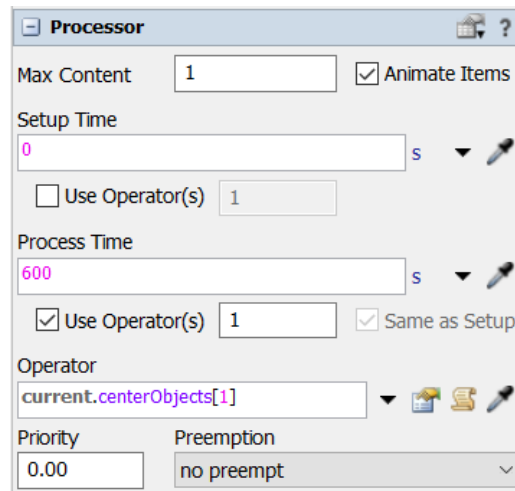


Figure 8

For the **kneading**, the white box will not be clicked and, as a consequence, the parameters will look like this.

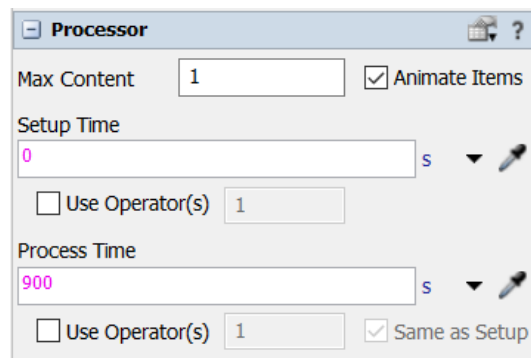


Figure 9

Finally, similarly to what happened with the source, the processors will need to be interconnect by means of output-input port connections, so that the flow items have a path to follow. So far, these connections will be joining the source with processor 1, processor 1 with processor 2, and so on and so forth, until processor 4 is reached. It will not be mentioned in further elements in the model, but all the fixed resources will be joined by means of output-input port connexions, so that the route for the flow items is completed. Also, whenever a fixed resource requires de use of a task executer, it will imply a central port connexion has been created between the two elements.

- 3) At this point in the process, it is important to establish what is the goal to be achieved by the production of 1 complete batch. That is, how many bread buns can be produced. According to the available data of the real system, that has been used as the basis for the construction of this model, the bread production system produces **1440 buns** per batch. The way this 1440 buns are obtained is quite simple: each tray has room for 20 buns, located in 4 rows of 5 elements each. A trolley can accommodate up to 18 trays and, knowing 1440 buns are to be achieved, 4 full trolleys will be needed to fulfil a

complete batch. Previously, it was being assumed that 4 trolleys were used per batch, but now it has been explained why.

$$\mathbf{Buns\ per\ batch} = 20 \frac{\mathit{buns}}{\mathit{tray}} \cdot 18 \frac{\mathit{trays}}{\mathit{trolley}} \cdot 4 \frac{\mathit{trolleys}}{\mathit{batch}} = \mathbf{1440} \frac{\mathit{buns}}{\mathit{batch}}$$

So, now it will become a matter of how 1 dough, being represented by a flow item, can be divided into 1440 buns in a sequential mode, according to the stages defining the complete process. Not only this, after the division, buns will need to be packed in an organised way so that the fermentation, baking, and cooling stages are performed according to the plan and, eventually, the unpacking of the bread forming the trays inside the trolleys will be done, once the cooling process is over. This way, the bread buns will be ready to be delivered to the customers.

- 4) To start the dividing of the dough, the first step is to complete a **1st division**. To separate elements, what is required in order divide the dough, a processor is not enough. As it was previously mentioned, a **separator** will be needed to perform this kind of tasks. Separators combine the ability of processors to process flow items, by means of providing a *Setup* and *Process Time* just like processors would do, with the differentiating feature of being able to separate them, by either the *Split* option, what divides a flow item into a certain number of new flow items, or by the *Unpack* one, what, in the case of a flow item being of the form of a *Pack* of several different flow items, allows the separation of the *Entire Contents*.

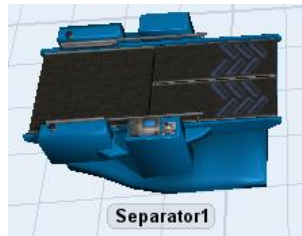


Figure 10

The *Split* option will apply in this case since what is intended is to divide the dough (1 flow item) into a certain number of divisions (n flow items). The data for the system being studied confirms that **48** divisions are to be obtained with the total dough in order to be able to meet the requirements of the production per batch.

Here, it becomes a little trickier to deal with the machines, since the combination of processing and separation might hinder the ability of the modeller to properly match the data for the time of the stage. Then, it is important to understand how a separator works in *FlexSim*: the flow item that enters the separator takes all the *Process Time* (plus the *Setup Time*, in case it exists) to get from the beginning to the end of the separator. This initial part is just like the case for a processor. It is just at the end of this processor activity within the separator when the separation takes place, instantaneously. That is, for the case being analysed, the divisions of the dough are not created progressively

from the beginning of the task to its end, what is known to be the case in the real system, they are created just before leaving the separator, and all the divisions at once.

Knowing that this 1st division takes 11 minutes, it would correspond to a time per division, in the real system, of

$$\frac{11 \text{ minutes} \cdot \frac{60 \text{ seconds}}{1 \text{ minute}}}{48 \text{ divisions}} = 13.75 \frac{\text{seconds}}{\text{division}}$$

However, this time per division cannot be introduced in *FlexSim* since it would yield a wrong result. The *Process Time* refers to the **total time** it takes to carry out the complete task, and not how much time it is required for a single division. Therefore, in the processor parameters of the separator, it will be written the total time in seconds, that is, **660** seconds.

Regarding the separation itself, it will be specified that, within the *Separate Mode: Split*, the target *Quantity* must be 48.

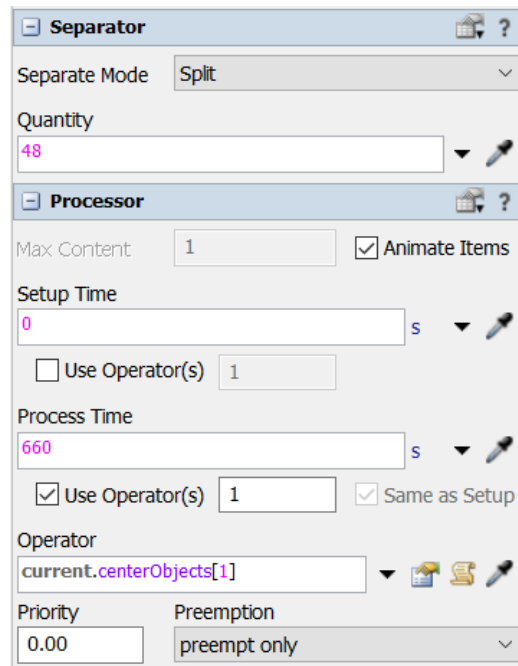


Figure 11

Unlike the case of the processors explained in 2), the *Preemption* option of the separator reads *preempt only* instead of *no preempt*. As it was briefly introduced, *Preemption* has to do with the order in which a task executor performs the different it has been commanded. More specifically, *Preemption* refers to the interruption of active tasks being carried out by task executors to work on a pre-empting task and determines what happens to the active task being interrupted. It will not be explained in detail the reasons why a task in the model being built is pre-empted or not, it is just important to understand that, for a correct imitation of the real system, some activities will need to be pre-empted. So, for the stage concerning this separator, it will be necessary to *preempt only* so that the operator being called for the *Process Time* behaves in a way

that properly represents the way a real worker would behave in the real production system.

- 5) Next, the 48 divisions of the dough obtained in the 1st division have to be shaped, before they continue with their progressive division process. This shaping will be modelled by means of a processor, but prior to entering the processor and as an intermediate step between the separator and the processor, a **queue** will be utilised as a means of temporary storage.

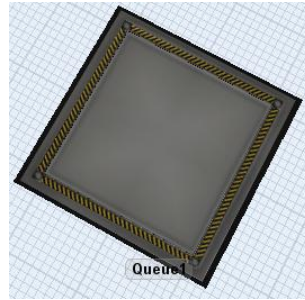


Figure 12

The reason why the use of a queue makes sense is that it allows the modeller to visually represent the place where the divisions obtained in the previous task would be “stored”, waiting for the worker to start shaping them once the previous task is over. Moreover, its parameters allow to define the maximum number of divisions that can be present within it (*Max Content*) and they also allow to perform batches (*Perform Batching*) that guarantee the correct number of flow items that will go on to the following element in the process flow, a processor to model the shaping of the divisions in this case.

So, it will be performed a batch with a size of 48. This way, the queue will only release its content whenever the batch is finished: in this case, the 48 divisions are created by the separator as soon as its *Process Time* finishes so, at that moment, the 48 divisions will instantaneously fill the batch in the queue. It will be seen later that this is not always the case since batches will normally be filled progressively. Queue batching is a smart way of ensuring that the activities are always performed the same way, following documented procedures so that, in the end, a successful 1440 buns of bread per batch is always obtained.

Here, because of the language, the term **batch** can be misleading: firstly, it was introduced to refer to a complete unit of production, composed of 1440 buns as it was explained previously. However, it is being used now to express a way of gathering flow items together inside a queue. So, it will be important to understand at any time what batch the text is referring to.

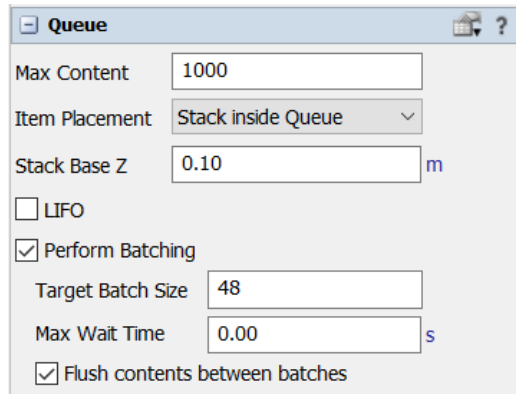


Figure 13

When batching is performed, the maximum content of the queue is no longer relevant because it will not be reached. However, it is critical because if the *Max Content* is smaller than the *Target Batch Size*, this second will not be reached and, therefore, the queue will not be allowed to release the items downstream. The box beside *Flush contents between batches* is very important as well, since it prevents new flow items to enter the queue until all the elements of the active batch have left. Otherwise, it would be impossible to determine when a complete batch is released. Finally, by including a *Max Wait Time* of 0 seconds it is implied that no maximum wait time is imposed to the queue. If a *Max Wait Time* were imposed it would mean that, regardless of not meeting the *Target Batch Size* by that time, the queue would be forced to release the items.

- 6) Right after the queue, a **processor** will model the **shaping of the 1st division**. This processor represents how a worker shapes the 48 divisions stored at the queue by hand, so this task will take a certain amount of time per division. In this case, it is necessary to include this time as the *Process Time* of the processor, and not the total time of the task. This makes sense since, because of how the model is being built, the processor will be receiving 48 items, representing the divisions, one at a time. If the total time were introduced in the processor's parameters, the time it would take to process the complete batch of 48 divisions would be equal to 48 times the real one. Therefore, it seems reasonable to consider as the *Process Time* for this case the following.

$$\frac{14 \text{ minutes} \cdot \frac{60 \text{ seconds}}{1 \text{ minute}}}{48 \text{ divisions}} = 17.5 \frac{\text{seconds}}{\text{division}}$$

Employing these 17.5 seconds, it is fulfilled that the completion of the task takes exactly 840 seconds, or 14 minutes, what coincides with what was indicated by the data of the real system. There is no *Setup Time*, and the processor will need to use a task executer for its *Process Time*.

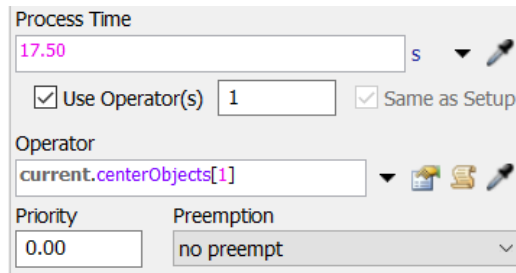


Figure 14

- 7) After the shaping of the 1st division, the dough undergoes the 2nd division. Similarly to how the 1st division was represented, a **separator** will be used in the representation of the 2nd division. A balling machine is employed in this activity so that, with the help of a worker, easy-to-use dough balls are obtained, and that allows the master baker to begin with the shaping process that starts right after this last division of the dough. The balling machine takes 1 shaped 1st division and produces 3 rounded and smooth balls. Then, it is to assume that 144 dough balls will be created through this 2nd division.

$$48 \text{ divisions} \cdot 3 \frac{\text{balls}}{\text{division}} = 144 \text{ balls}$$

Again, something similar to what happened with the *Process Time* of the processor in 6), the total time of the process has been given, and not the time it takes the balling machine to perform 3 balls. Hence, the *Process Time* to be introduced in the parameters must be equal to the total time divided by 48, this way after the 48 shaped balls have been both processed and *Split* by the separator, the total time of the task will match the data provided for the real case. The complete stage takes 26 minutes, so

$$\frac{26 \text{ minutes} \cdot \frac{60 \text{ seconds}}{1 \text{ minute}}}{48 \text{ shaped divisions}} = 32.5 \frac{\text{seconds}}{\text{shaped division}}$$

No *Setup Time* for the task and an operator required.

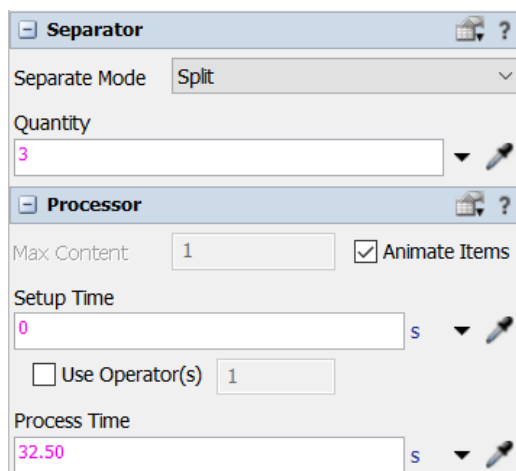
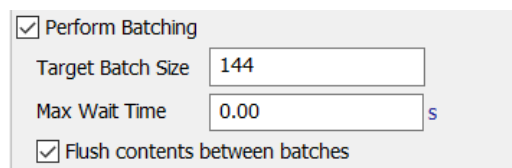


Figure 15

- 8) Now, it is time for the **shaping**. The complete stage will be composed of three parts, that will be performed sequentially. To be able to do this it will be of great importance the pre-emption of activities that was discussed previously. Not using the appropriate commands would produce a model that might not replicate the real behaviour of the system properly. For the sake of simplicity and because of the lack of specific data related to these internal parts of the shaping stage, it has been chosen to divide the total time in three equal parts to represent the time it takes to fulfil each of the three parts. That is, each part will take a total **40 minutes** to be completed. Knowing that 4 trolleys will eventually leave the shaping stage with the total 1440 buns after the 120 minutes, it is easy to see that the way of proceeding is the following: it takes 40 minutes to perform the activities that correspond to 4 trolleys, that is, 10 minutes are used per trolley. Then, after the 10 minutes of the first part, the 10 minutes of the second come and, finally, the 10 minutes of the third. That makes 30 minutes per trolley, what, for a total 4 trolleys, yields 120 minutes for the complete process indeed.

The **first part** is the one that really deals with the **shaping of bread**. From the previous stage, it is known that 144 dough balls are being delivered so that a master baker starts creating the bread buns. Similarly to what happened before entering the shaping of the 1st division, where a queue was included to perform a batch of 48 flow items, another **queue** will be introduced before the entry of the shaping stage. Here, the performed batch will need to have a *Target Batch Size* of 144 items.



<input checked="" type="checkbox"/> Perform Batching
Target Batch Size <input type="text" value="144"/>
Max Wait Time <input type="text" value="0.00"/> s
<input checked="" type="checkbox"/> Flush contents between batches

Figure 16

It is assumed that the worker that is performing the 2nd division with the balling machine will have available room to deliver 4 complete batches of 144 dough balls. That is, 4 equal queues with the previous parameters will be used. The fact that this available room is needed has to do with the time difference between the tasks prior to the shaping and the shaping itself. If just one queue is added, by the time these tasks prior to the shaping in the second batch are finished, the shaping will still be happening, what will cause that the 2nd division will have to wait for the queue to release the total 144 items' batch before letting any more items in (something that was imposed by clicking the box beside *Flush contents between batches*). With 4 queues it is ensured that this unnecessary wait does not happen. The reason why 4 queues are enough does not matter now, but it could be easily proved by means of a tool that will be introduced and used later, the *Experimenter*.

After the queue, it will be necessary to turn the 144 dough balls into 1440 bread buns, for what it will be necessary to use another **separator**. The logic to follow in this case is the same as it was for the separator in 7), with the differences of needing to *Split* the flow items into 10 (this way, 10 divisions of 144 balls provides 1440 buns), instead of 3,

and with a different *Process Time*. Here it is important to understand what total time is to be achieved by the process, which in this case, and according to what has been explained above, must be **10 minutes**, or 600 seconds.

So, as 1 trolley is able to accommodate a total of 360 bread buns, what can be derived by multiplying 20 buns per tray times 18 trays per trolley or, equivalently, by dividing the total 1440 buns by 4, 36 balls will be needed to complete 1 trolley. That means, processing these 36 balls by the separator in this first part of the shaping must result in a total 10 minutes, that is

$$\frac{10 \text{ minutes} \cdot \frac{60 \text{ seconds}}{1 \text{ minute}}}{36 \text{ balls}} = 16.67 \frac{\text{seconds}}{\text{ball}}$$

Again, no Setup Time is specified, and a task executor will be called from the separator.

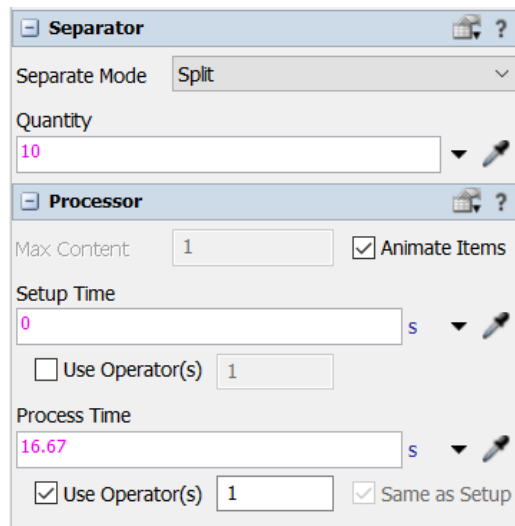


Figure 17

The **second part** is devoted to the filling of trays with the already shaped bread buns. For the completion of this task, another type of fixed resource will be introduced: the **combiner**. It could be said that combiners have the same features as separators, but inverted. They can combine flow items by either packing (*Pack*), joining (*Join*) or batching them (*Batch*).

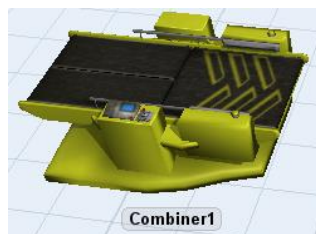


Figure 18

For this case, the *Pack* option will be used since the target of this combiner is to obtain a tray full of buns, what can be represented by a pack of 1 tray and 20 bread buns. What makes combiners special is precisely the way in which the combining action is carried

out: by construction, a combiner is allowed to combine flow items coming in through different input ports (*Input Port 1, Input Port 2...*). However, its configuration establishes by default that the *Input Port 1* will only provide 1 item to the combination, so the only parameters to deal with will be the number of items coming from each of the remaining input ports that will be combined with the item coming from *Input Port 1*.

So, as a pack will have 1 tray and 20 buns, it is necessary to add a **source** to create the trays and connect it to the *Input Port 1* of the combiner. By connecting the output port of the separator from the previous part to the *Input Port 2*, it will be possible to determine how many buns are required to be packed with 1 tray. *FlexSim* defines the element being 1 by default as the *Container* in the combination, whereas the other elements that are added to the *Container* are defined as *Contents*.

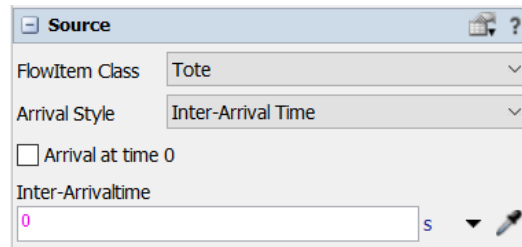


Figure 19

It has been chosen a *FlowItem Class* different from a box (*Tote*), so that the pack is clearly seen and the *Inter-Arrivaltime* being equal to 0 implies trays are always available for the formation of the packs.

For the *Process Time*, the logic that applies is the following: in the combiner, this time refers to the time it takes to produce a complete combination, in this case, a pack representing a tray full of buns. Knowing that 360 buns leave the first part of the shaping stage, 18 is the number of trays that will be filled with this amount of bread buns, since 360 buns divided by 20 buns per tray gives 18 trays. So, the time to be introduced in the parameters is

$$\frac{10 \text{ minutes} \cdot \frac{60 \text{ seconds}}{1 \text{ minute}}}{18 \text{ trays}} = 33.33 \frac{\text{seconds}}{\text{tray}}$$

No *Setup Time* is specified, and a task executer will be needed.

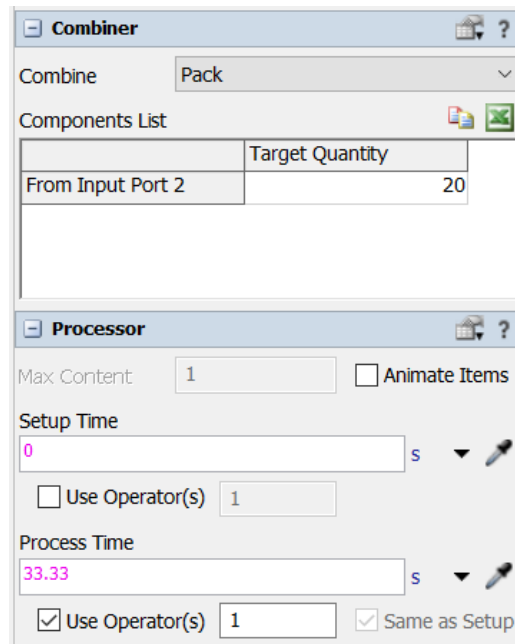


Figure 20

Finally, the **third part** will represent the filling of the trolleys with full trays. So, another **combiner** will be required for this last part of the shaping stage.

It can be easily understood that the reasoning for this part is pretty similar to the one for the previous: in the second part, 1 tray was to be combined with 20 buns in order to complete a pack that would leave the combiner after the defined *Process Time*. Now, in the third part, 1 trolley is to be combined with 18 trays in order to complete a pack that will be ready to leave the combiner and the shaping stage after its corresponding *Process Time*, since this third part is the last one of the shaping process.

Of course, as it happened in the second part, a **source** is needed to create the flow items that represent the empty trolleys. Again, a new *FlowItem Class* is chosen (Pallet), and the *Inter-Arrivaltime* is kept equal to 0.

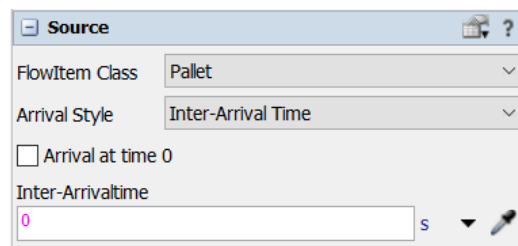


Figure 21

By connecting this source to the *Input Port 1* of the combiner, the pack can be easily obtained by indicating how many trays must be combined with 1 trolley. It was previously explained that this 1 trolley is set by default when connecting the source creating them to the *Input Port 1*. This way, the number of components from *Input Port 1*, which always equals 1 (*Container*), is not included in the table of parameters of the combiner, unlike the number of components from input ports other than *Input Port 1*,

whose value can be introduced and might be different from 1 (*Contents*). In this case, as 1 full trolley contains 18 trays, the number of components from *Input Port 2* must be 18.

According to what was explained previously, the time per part and trolley must be of 10 minutes (so that 3 parts of 10 minutes times 4 trolleys complete the total 120 minutes). In this case, as the pack being created in the combiner represents 1 full trolley, it means the *Process Time* to be introduced is precisely these 10 minutes, or **600** seconds (per trolley). Again, no *Setup time* is mentioned, and a task executer will be called from the combiner.

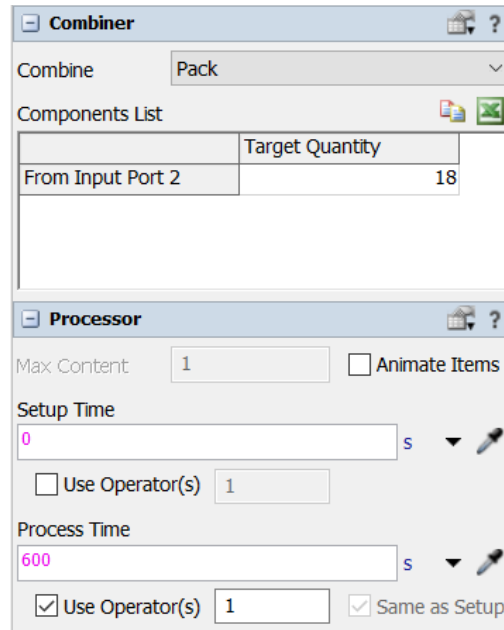


Figure 22

- 9) After the shaping stage, it is time to go on with the process, what leads to the **fermentation**. Once the bread has spent enough time inside the fermentation chamber, it will be carried to the ovens for **baking** and, right after that, it will undergo the **cooling** process. These three stages can be similarly represented in *FlexSim* because they are all based on a *Process Time* that is applied to each of the trolleys that leave the shaping stage. Regarding the available room for each of these activities, what will allow to process more than one trolley at a time, according to the given data for the real system, it is known that: the fermentation chamber can accommodate up to **4** trolleys at the same time, there are 2 ovens with a capacity of 1 trolley each, so **2** trolleys can be baked at the same time, and there is 1 fan that can cool **2** trolleys at the same time.

Then, the activities are represented by means of **processors** and, to represent all the available room for trolleys to be processed, more than 1 processor of the same characteristics will be included in parallel. That is, at the exit of the combiner from the last part of the shaping, 4 output-input port connexions will be created to allow the flow items, in this case the trolleys, to get access to 1 of the 4 empty spots inside the fermentation chamber, what is modelled with 4 parallel processors. For the baking, the logic is similar: 2 parallel processors instead of 4 at the exit of the fermentation chamber.

The same applies for the cooling: 2 parallel processors right after the 2 ovens. It is important that the output-input port connexions are created so that any trolley leaving a process can access any available spot in the following process. To illustrate this, an image of how the connexions should be performed looks like this.

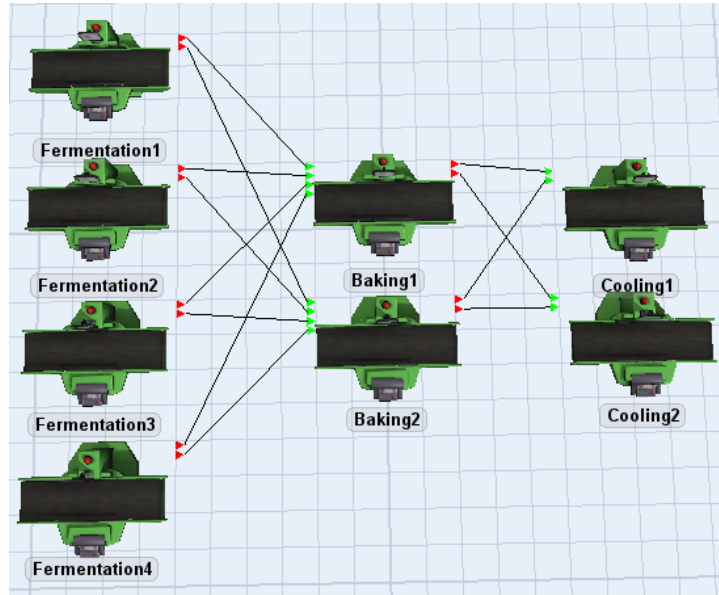


Figure 23

Once the connexions are completed properly, it is a matter of defining the *Process Time* of each of the three groups of processors. **Fermentation** takes 30 minutes, or **1800** seconds; **baking** takes 15 minutes and so does **cooling**, that is, **900** seconds.

One remarkable feature of these three activities is that they do not require the presence of an operator to perform the tasks together with the machinery. Then, the *Use Operator(s)* white box for the *Process Time* will not be clicked for these three cases. On the contrary, despite not requiring the operators during the completion of the tasks, the processors do need a task executer to transport the flow items form one processor to another. This will be considered to replicate the reality as accurately as possible, since no times have been provided by the data from the real system for this intermediate transports. However, if these transports were considered as non-existent by the modeller it would not make sense from the real world's perspective. For example, according to the real system data, a trolley would take $30+15+15=60$ minutes to complete these three stages. Nevertheless, it seems obvious that some extra time should be added, representing the time it takes the workers to carry the trolley from one step to the following. In the analysis of the results, transport will be taken into account as a relevant factor for evaluating how efficient the process is.

Therefore, the white box beside *Use Transport* will be now clicked. It is seen that this transport refers to the output of the fixed resource, that is, whenever the activity is finished a task executer is called to carry the flow item from the current element to the following. So, since it is stated that workers will transport the trolleys to all the discussed 3 stages and, it could also be assumed, from the cooling once it is finished to the packing

stage, the *Use Transport* box must be clicked: the combiner prior to the fermentation, the processors in fermentation, the processors in baking and the processors in cooling. As the case for operators required for the *Process Time* of certain tasks, the calling for transport is performed by means of a central port connexion between the fixed resource calling and the task executer.

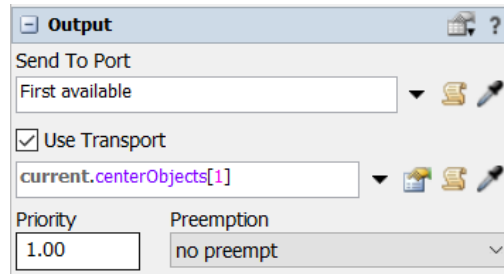


Figure 24

As it was mentioned earlier, *Priority* levels will determine the order in which task executers performed their assigned activities. The case of the image corresponds to the output of the fermentation processors. This number 1, being bigger than the default number, which is 0, implies that activities being numbered 1 *Priority* are “more important” than the ones being numbered 0, allowing the overall process to be completed in a sensible manner, so that the model represents the real system appropriately.

Also, it could happen, as in the case of the combiner prior to the fermentation stage, that a central port connexion is already being used for calling a task executer to perform the activity. So, in case the transport is meant to be done by another task executer, another central port connexion must be created (*centerObjects[2]*).

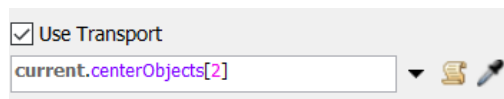


Figure 25

- 10) Finally, once the baked bread leaves the cooling process, it is ready for its **packing** and delivery according to the production order. Since no information has been given about this production order, which could vary from time to time, it will be assumed that all the bread buns are “packed” by reaching a **sink**. This sink represents the different boxes where the different quantities of buns would be introduced so that a specific production order is met. It will be considered that the time it would take to perform the real task is equivalent to the time it would take to perform this packing into a single sink, what could be seen as an enormous box where all the produced buns are introduced. For instance, if a production order must include 10 packs of 144 buns each, what can be met by 1 batch of 1440 buns, the sink represents an area where 10 empty boxes wait for the arrival of the bread. Then, an operator is in charge of filling each of the boxes with 144 bread buns. By using this sink, any combination of boxes could be conforming the production order, but the representation will not vary.

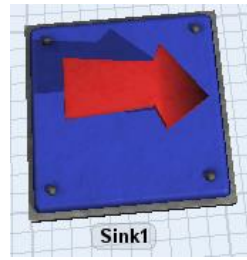


Figure 26

But first, in order to get to the final sink where all the bread buns will be deposited, the combinations that allowed buns being put on trays and trays being put on trolleys must be undone. As it was mentioned when the combiner was introduced in 8), it was said it has the same features as separators, but inverted. So, if the process that allowed to put bread buns in trolleys required 2 consecutive combinations, performed by combiners, the process of releasing the buns from the trolleys will require 2 consecutive separations, performed by **separators**.

Unlike the previous cases that were using a separator, whose goal was to divide the dough by splitting the flow items into a certain number of divisions (Split), in this case, the intended separation will look for the unpacking of previously packed combinations, for what *Separate Mode: Unpack* will be used.

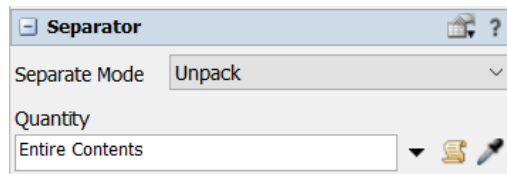


Figure 27

This option must release *Entire Contents* through the output ports of the separator. The output default option for the separator releases the *Container* through output port 1 and the *Contents* through output port 2.

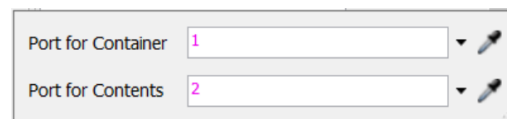


Figure 28

For the first separation, the trolley is the *Container* and the trays with the buns are the *Contents*. So, as the trays with the bread need to go on with the unpacking, the output port 2 will be connected to an input port in a new separator, whereas the output port 1 will be connected to a sink for trolleys. Once the second separator is reached, and similarly to how it was proceeded with the first, the *Container*, in this case the tray, will leave the separator through output port 1, and the *Contents*, in this case the bread buns, will leave the separator through output port 2. Another sink will be used for the trays and the already mentioned “general” sink, what represents the packing itself no matter what the production order is, will be waiting at the exit of output port 2, to receive the produced bread. The 3D layout will look like this.

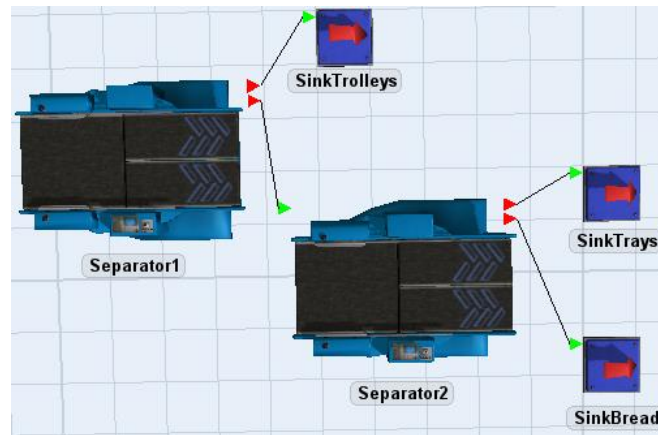


Figure 29

According to the given data, the process of packing a trolley that leaves the cooling stage takes 2 minutes, or **120** seconds. So, this time must be enough for the 2 separators to carry out their tasks appropriately. It has been considered that the first separator will use 30 seconds and that the second one will use the remaining 90 seconds. The first separator unpacks 1 combination per trolley since the *Container* for this case is the trolley and only 1 trolley is entering from the cooling stage at a time. Therefore, the *Process Time* in this first separator that, as it was seen previously, precedes the separation itself, must be equal to the total **30** seconds devoted to this first unpacking, because only 1 separation will occur. On the other hand, the second separator will not be performing just 1 separation, in which case the *Process Time* to introduce should be 90 seconds. It will perform as many divisions as *Containers* reach the separator. So, for this second separator, the *Container* is the tray and a total of 18 trays per trolley will be reaching this separator once they have been unpacked from the trolley-trays combination. Then, the processing of the 18 trays must take 90 seconds, that is, **5** seconds per tray and the value for the *Process Time*.

Again, no *Setup Time*, a task executor required for the *Process Time* and some specific *Priority* and *Preemption*. Next figure represents the parameter for the second separator, that are exactly the same as the ones for the first separator, except for the *Process Time* being 5 instead of 30 seconds.

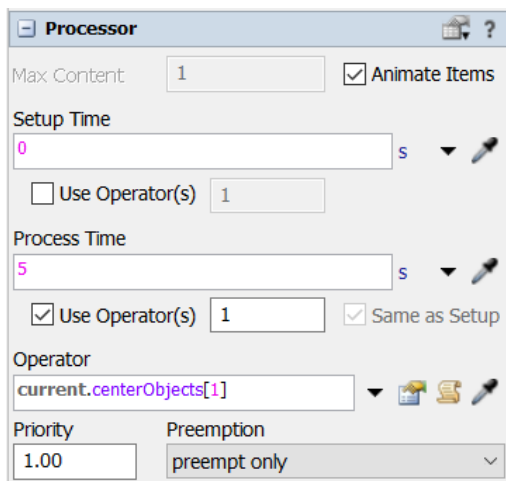


Figure 30

By means of these 10 steps, it has been possible to build this introductory model in a satisfactory way. Now, it will be checked if the results obtained by means of the theoretical system can be obtained by running the model. In case the results coincide, or are reasonably similar, the model will have proved to be adequate.

4.2.1 Verification and validation of the introductory theoretical model

It was imposed by the resolution of the theoretical case that, in order to keep up with the rhythm of production of 1 batch every 120 minutes after the first batch has been produced, they were needed 2 operators for the complete process. So, these workers will be connected through central ports connexions to the fixed resources that will be using each of them.

Knowing this, it is time now to check the results of the running of the model. Firstly, regarding the time it takes to produce the first batch, and secondly, regarding the cycle time of 120 minutes.

To run just the production of a batch, just a single flow item must enter the process through the initial source, named *RawMaterials*. To achieve this, *Arrival Sequence* is chosen as the *Arrival Style* and the table is edited so that there is only 1 arrival at time 0, what happens by default.

	ItemName	Quantity
Arrival1	Product	1

Figure 31

By running the model, the *Run Time* stops at 16601.75 seconds, what implies that all the flow items have reached their corresponding sinks, and there are no more items still inside the process.



Figure 32

In order to make sure this time actually represents the total time it takes the first batch to be produced, it can be checked the input of the last sink, the one receiving all the produced bread buns, which has been named *SinkBread*. If this value is equal to 1440, the obtained time does correspond to a full first batch.

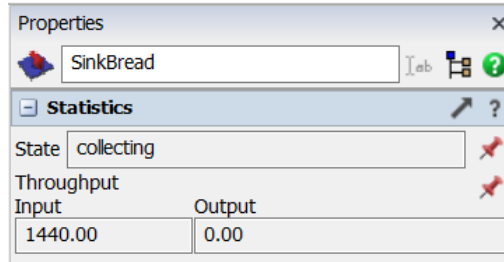


Figure 33

Then, knowing that the 16601.75 seconds correspond to the completion of the first batch, it corresponds to a total 276.7 minutes approximately. The theoretical solution was **271.5** minutes. It might already seem to be quite close but, there is still one consideration to be made. It was mentioned that transport at certain stages of the overall process was not taken into account when gathering the data for the real process, and that is precisely why these two times do not coincide. If a *Dashboard* is created, which is a tool that *FlexSim* uses to show different graphs and statistics for the model being run, it can be analysed the *State* of the workers throughout the process, that is, if they are being used, idle, or travelling, amongst others. There are different possibilities to present this information, but an interesting one is the *Pie Chart*.

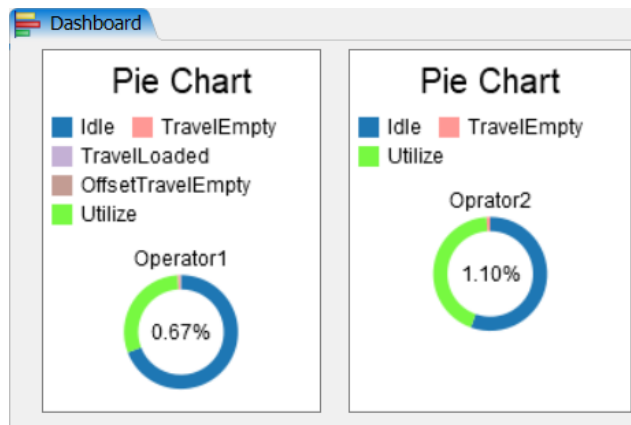


Figure 34

During the percentage of the total time written inside the circles, the operators have been under *TravelEmpty*. Then, it is reasonable to assume that this time was not considered by the theoretical problem, and that it may be the difference between the two different times obtained. If this total 1.77% of the time of the model was removed, that is, just using the 98.23% of the 16601.75 seconds for the process itself, the time that is obtained is around 16307.9 seconds, or **271.793** minutes. This value is quite close to the theoretical one so, apparently, the model is replicating the real system quite well.

Secondly, it will be checked if the cycle time matches the 120 minutes that the bottleneck activity takes to be completed. Also, the working time of both operators must match the theoretical values. As a theoretical calculation, 11 batches will be produced by the simulation of the model. The first of them must take 271.5 minutes, but the remaining 10 must take 120 minutes each. Of course, this shift for 2 workers would not be possible, since more than 24 hours are expected to be the result of the simulation. It is just a theoretical calculation to ensure that the results of the simulation match the theoretical results of the ALBP.

To run the simulation concerning these 11 batches, it is enough to modify the *Arrival Style* of the source *RawMaterials*. The option *Arrival Schedule* will be chosen for this case, and the times for the 11 arrivals must be introduced in the table. The first arrival occurs at time 0, just like the previous case; the second, 89.5 minutes after the first arrival. This way, when the flow items enter the shaping stage, another batch is getting ready so that by the time the shaping finishes it can start with the following batch. The third and all the remaining arrivals will happen 120 minutes after the previous one occurred. Knowing that 89.5 minutes are 5370 seconds, and 120 minutes are 7200 seconds, the complete table for the *Arrival Schedule* looks like this.

	ArrivalTime	ItemName	Quantity
Arrival1	0	Product	1
Arrival2	5370	Product	1
Arrival3	12570	Product	1
Arrival4	19770	Product	1
Arrival5	26970	Product	1
Arrival6	34170	Product	1
Arrival7	41370	Product	1
Arrival8	48570	Product	1
Arrival9	55770	Product	1
Arrival10	62970	Product	1
Arrival11	70170	Product	1

Figure 35

Once the model is run, it can be seen that the rhythm of production is kept with no interferences, unexpected bottlenecks, or any kind of problems. It seems that the model is providing a good representation of the real system.

For example, it can be checked if the time employed by worker in the shaping stage matches with the theoretical one. Knowing that 11 batches have been completed, this worker must have used 1320 minutes, or 79200 seconds. To do this, another *Pie Chart* will be used in the *Dashboard* (actually, it will be the same as the one for *Operator2* in *Figure 31*, but that has been modified by the fact the problem being analysed has changed), and this time, the *Utilize* percentage will be checked for *Operator2*. This value, multiplying the *Run Time* at which the simulation has finished, should match the theoretical 79200 seconds.

Run Time: 98548.44

Figure 36

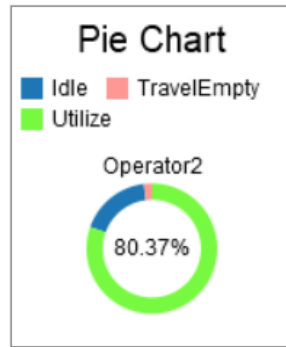


Figure 37

So, multiplying 98548.44 seconds times 0.8037, the time the *Operator2* is being used equals **79203.38** seconds, which basically the same as the theoretical value that was expected.

Also, it could be checked how much time the worker performing the remainder of the activities has spent doing them in the simulated model. According to the theoretical results, this worker was used during 82.5 minutes per batch, that is, a total of 907.5 minutes, or 54450 seconds, in the 11 batches.

Multiplying the same *Run Time* as above times the percentage indicating *Utilize* for *Operator1* in the following Pie Chart, the simulation result is obtained.

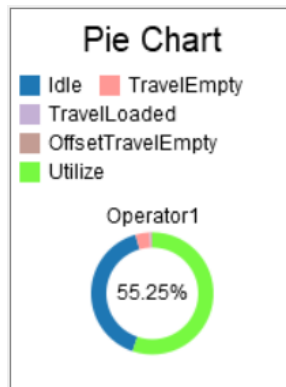


Figure 38

Then, the multiplication provides a result of **54448.01** seconds. Again, the result obtained by means of the model simulation is practically the same as the theoretical value.

After analysing the similarities between the theoretical results and the ones obtained by means of running the simulation model, it becomes quite clear that the model is indeed replicating the real system it represents appropriately, suggesting that the building of the model has been successful so far. This is a very important achievement in order to successfully continue and, eventually, finish with the overall simulation study. This model, as it was previously mentioned, has been used as an introductory model, but it will certainly be the basis for the construction of the model of the real system. It will be seen that not too many features of the next model differ from the ones that have been analysed for the current one, so just slight changes will be applied to this introductory model, which has already proven to be successful.

4.3 BUILDING OF THE MODEL FOR THE REAL PRODUCTION SYSTEM

It is important to understand why the previous model was not a completely accurate representation of the real system, what is mainly exemplified by the fact that all the data and the variables for the machinery and the workers in the different tasks within the system are **deterministic**. Of course, for the resolution of the theoretical case, this circumstance has made things easier but, it is not strictly realistic. The data regarding the divisions of the dough, amounts inside the trays and trolleys, etc, is deterministic by definition, and so it will remain deterministic for the new model. However, the time data that has been provided for the duration of the activities is just the average value. The given value is the one being the most likely to happen, but it does not imply that no other values different from this mean value could occur, which will certainly be the case. That is, time related data and variables are **stochastic** by nature.

Moreover, for the sake of simplicity in the calculation of the theoretical case, the average duration of some activities was slightly modified from the one provided by the original source of the data [25], in order to deal with more exact values. The values in the following table are the ones that were modified. The capital letters that are used to represent the tasks correspond to the assignation that was performed in *Table 2*. The activities that are not included will keep the same average *Process Time* for the real case as the one they had for the theoretical calculation.

Task	A	B	D	E	F	G	H (part 2)	H (part 3)
Average <i>Process Time</i> theoretical case (s)	600	720	90	660	17.5	32.5	33.33	600
Average <i>Process Time</i> real case (s)	587	746	86	662	16.8	32.4	33.34	600.03

Table 5

Despite knowing that the reality is stochastic, the initial simulation for the real model will be performed considering deterministic variables only. That is, using the average *Process Time* for the different tasks in the real case, which are either stated above or maintained from the theoretical case (for the activities that are not included in *Table 5*). Again, this is not realistic, but the fact that mean values are being used should provide a successful solution of the simulation. If bad results were obtained when running this model, it would be clear that something is wrong.

After the completion of this initial approach with the deterministic model, the stochastic model will be built. For that, appropriate statistical distributions will be assigned to the different activities so that the closest to the behaviour of the real system can be modelled. It will be discussed later how the choice for the most suitable statistical distributions can be made.

A comparative analysis will be carried out in order to check the similarities and the differences between the two models, and it will eventually be possible to determine if the model is valid and appropriate for the simulation of the real production system, based on the real data provided regarding working hours and total production in the company under study [25, tab. 2.3 and 3.14].

4.3.1 Deterministic model

To start building this model, the one for the theoretical case will serve as the initial point, as it was previously mentioned. The first modification will be to include the average *Process Time* for the real case to their corresponding tasks.

Secondly, it is important to understand that the task assignment that was implemented for the theoretical case does not apply for the real case. The assignment just considered 2 operators working in series, in a single production line and assigned to specific tasks. However, it will be explained that these assumptions for the theoretical case were not representing exactly the features of the real system. They just served for the theoretical approach.

For example, the fact that the theoretical case considered 1 single production line does not accurately represent the reality, since it is known that, at certain stages, it is possible to work in parallel. This single production line was split in the theoretical model when the fermentation process arrived, where several parallel processors were required. However, until that stage, and because of the characteristics of the tasks, the line did not consider the possibility of working in parallel since, for the case being studied, 1 line was enough.

There are 2 kneading machines available and, at the shaping area, there is room for 2 master bakers to be able to shape bread at the same time. To duplicate the mentioned stages, which are the **kneading**, with the **extraction of the dough** as a consequence, and the three parts of the **shaping**, what includes the queue located at the entry, it can be easily done in *FlexSim*, by means of clicking at the fixed resource that is wanted to be duplicated and just copy-pasting it directly in the 3D layout. It will also be necessary to create the new output-input and central port connexions that are needed.

Since there exist now two available lines to process the dough balls coming from the 2nd division stage, it will be interesting to divide the work in 2 equal parts, what can be done by modifying the *Target Batch Size* of the 2 queues located at the entry of the 2 lines for the shaping stage, from 144 to **72**.

Before explaining how the task assignment must be modified for the model of the real case, it can be helpful to briefly introduce the concept of **standard time**. This term denotes the time it takes an average skilled worker, at a normal pace, to complete a task following a prescribed method. This standard time includes allowances for the operator to recover from fatigue and that also account for other incidental elements that might occur [32].

So, for the case of the real system being studied, all the data that has been provided corresponds to the standard time for each of the operations [25, tab. AVII.1, AVII.2, AVII.3 and AVII.4]. Of course, for the stages where machines perform on their own, the given time will not be a standard time but a machine time. Nevertheless, the two of them will be dealt with as equivalent. This implies that any worker, as long as they are qualified, can perform any task in the system. Therefore, for the building of the new model, no operators will be assigned to determined activities, but they will be assigned to all the activities at the same time. How is this possible?

FlexSim provides a tool that has not been discussed yet, since it did not apply for the case of the theoretical model, which is able to organise and distribute task executers that are connected to it to perform activities in fixed resources that are also connected to it. This tool, included in the *Task Executers* classification, is the **dispatcher**.

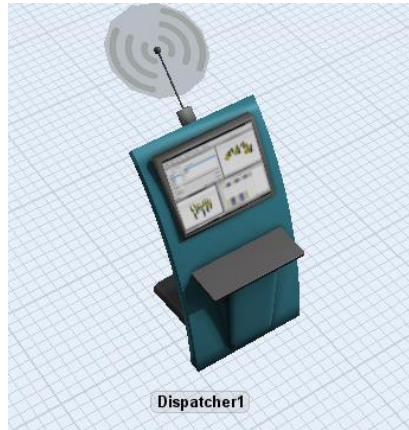


Figure 39

So, every activity that requires a task executer either for the process itself or for transport will call them by being connected to the dispatcher through a central port connexion. It is the duty of the dispatcher to decide which operator or any other task executer connected to it must perform each of the different activities. The group of task executers over whom the dispatcher has the control and the authority to assign the tasks will be connected to it through output-input connexions, starting at the dispatcher and finishing at each of the task executers. This is how the 3D layout for the 2 lines, with the dispatcher connected to all the activities that require its action. It would be left to add the work executers connected to the dispatcher, what will be done later.

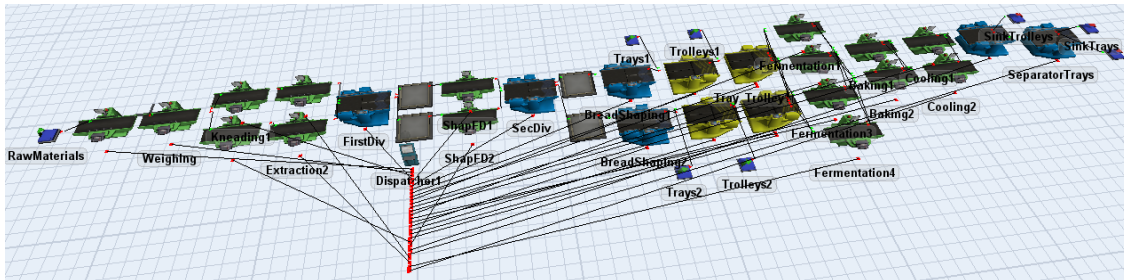


Figure 40

Now that the new layout is ready for the simulation of the real case and the dispatcher has been introduced, it is time to establish the conditions for the runs so that the model replicates the production of the real system appropriately. That is, it is necessary to determine the work shifts and the number of workers in each of the shifts. This is done according to the data provided.

Shift	Schedule	Number of workers
Morning	5:00-14:00	1
Afternoon	14:00-22:00	5
Night	21:00-5:00	4

Table 6 [25, tab. 2.1]

In [25, tab. AIX.1], it is stated that the number of hours of work in a month is equal to 2080. If a day is composed of the total hours the workers work in a day added together, which is equal to 9 h (1 operator, in the morning shift) + 40 h (5 operators, in the afternoon shift) + 32 h (4 operators, in the night shift) = 81 hours/day, it can be computed how many days of work fulfil the 2080 hours in 1 month.

$$\text{Days of work per month} = \frac{2080 \text{ hours/month}}{81 \text{ hours/day}} = 25.679 \text{ days/month}$$

To model the work shifts according to the daily schedule, it will be necessary to access the *Toolbox* of *FlexSim* and use the *Time Tables* tool. Using it, the correct working shifts can be attained by defining appropriate timetables for the workers to travel to a resting place, that would be modelling their home. That is, by ensuring the workers are resting at a certain time lapse it is achieved that they are working the remainder of the available time.

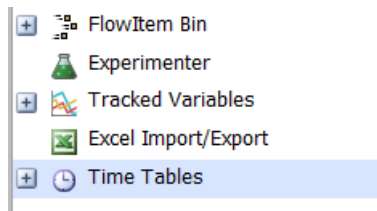


Figure 41

So, the 10 operators that are required during a day will be connected to the dispatcher in the 3D layout, and they will also be connected to a fixed resource that has not been discussed yet which will be modelling their place to rest. This resource is a *BasicFR*: this resource is basically designed so that it can be customised into a user library object. It presents a great deal of options but, for the case being represented, it will just serve as an object where the operators will be travelling to because of a *Down Function*.

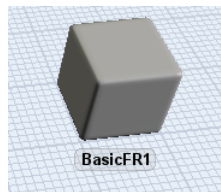


Figure 42

Three of them will be used to represent the “home” of the three different groups of workers, if they are divided by shifts: morning shift, HOME1; afternoon shift, HOME2; and night shift, HOME3.

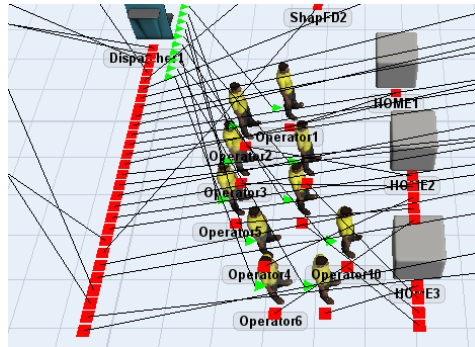


Figure 43

The process to build a *Time Table* is the following: firstly, the elements to which the *Time Table* is being applied must be introduced. Three *Time Tables* are needed for the three different groups, so the first will include *Operator1*; the second, *Operators 2, 3, 4, 5 and 6*; and the third, *Operators 7, 8, 9 and 10*. Here, it can be seen the case for the second *Time Table*.

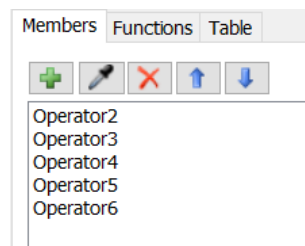


Figure 44

Secondly, in the tab *Functions* it is stated what the introduced elements must do when the *Down Function* and the *Resume Function* happen. Here, a specific *BasicFR* is assigned to each of the *Time Tables*. This can be done by selecting the second icon where a hand with a pointing index finger allows to edit the parameters. This is the case for the first of the *Time Tables*.

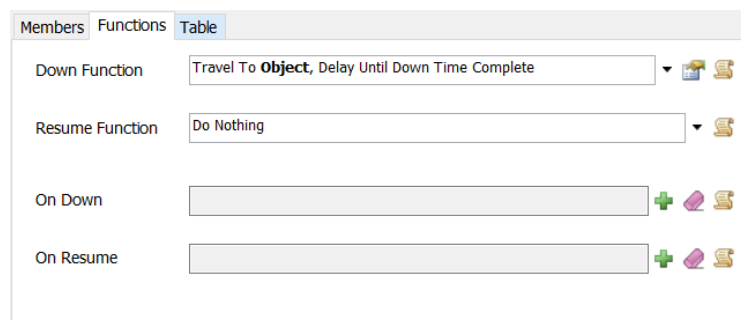


Figure 45

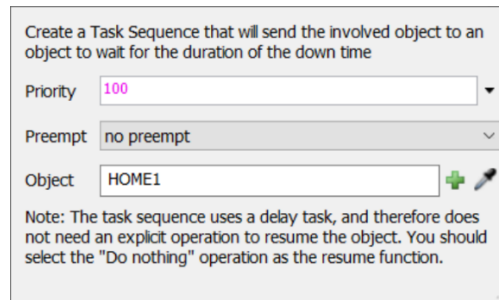


Figure 46

Finally, the last step to complete the building of a *Time Table* would be the definition of the resting time. For the morning shift, it must be achieved that the resting time goes from 14:00:00 to 5:00:00, in a way that the remaining time corresponds to the duration of the shift. For the afternoon shift, the resting time must go from 5:00:00 to 14:00:00 and from 22:00:00 to 5:00:00 (it is equivalent to saying that the resting time must go from 22:00:00 to 14:00:00, but it must be defined this way so that the first day of simulation also considers the resting time from 5:00:00 to 12:00:00). Eventually, for the night shift, the resting time must go from 5:00:00 to 21:00:00. It will be ordered that this schedule is repeated daily, for a total of 30 times. That is, the working schedule is obtained for a total of 30 days. The choice of this number 30 has been arbitrary, since the simulation time is going to be shorter than these 30 scheduled days, as it will be discussed later. As an example, it is shown the case for the night shift.

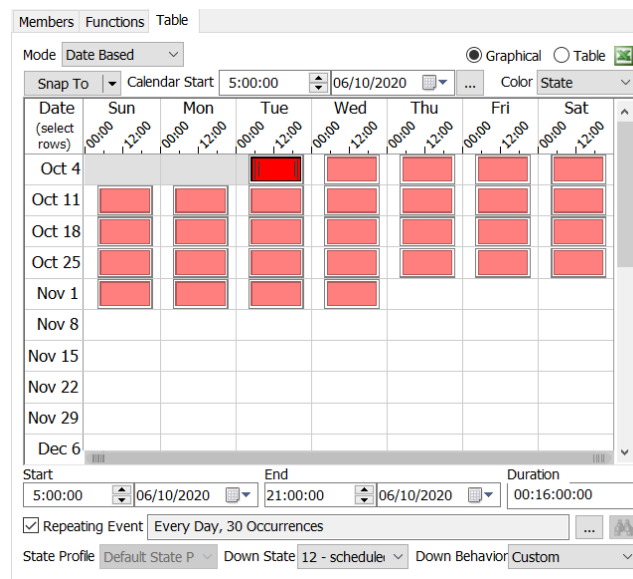


Figure 47

With the definition of the correct schedule, what includes the right duration of the shifts and number of workers in each of them, it is now possible to run the model accordingly. To do that, the *Experimenter* tool that was mentioned earlier will be of great help. It is accessed through *Statistics* and, as a matter of fact, the greatest value of this tool will be seen for the simulation of the stochastic model, but it will be used for the deterministic case as well.

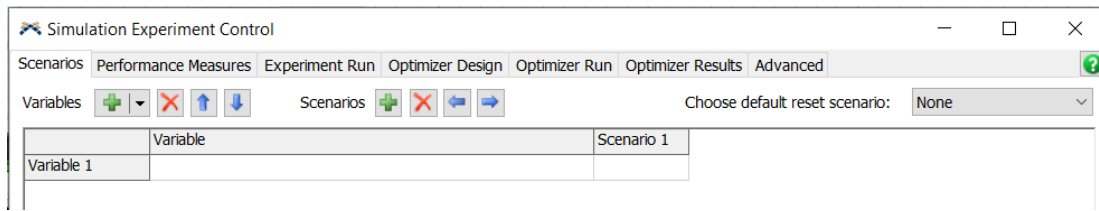


Figure 48

The *Experimenter* offers the possibility of analysing different cases (scenarios) for different variables, in order to obtain the performance measures of interest in a very effective and easy-to-handle way. By selecting the *Performance Measures* tab, it is possible to select the one that is wanted to be obtained with the analysis of the variables and scenarios. *Variables* whose modifications are wanted to be studied can be chosen, and different *Scenarios* to check the behaviour of the variable's change in the performance measure can be created. It can be understood that these relationships are no longer trivial and that this *Experimenter* tool, together with the *Optimizer* that is included within and that will be used later, really is an extremely helpful tool, which adds value to the entire simulation process. This unit could be considered as the core of the whole simulation software, since it is the truly powerful element that will enable the calculation, analysis and optimisation of the results. Without the *Experimenter*, it would still be possible to perform the simulation of different scenarios, but it would be much harder since it would be necessary to build all the different scenarios for each simulation. Instead, thanks to it, this task is straightforward. Moreover, the possibility of creating different replications per scenario is extremely useful, especially in the cases like the one that will be seen for the stochastic model.

From the beginning of the modelling process nothing was stated about the availability of raw materials and their rate of arrival to the system. Then, it has been assumed that there are no problems with the supply and, of course, it can be assumed that the workers will apply an *Inter-Arrivaltime* which is the most beneficial for them, normally representing a trade-off between long times, that might cause long periods of waiting, and short times, that can end up causing trouble because of the blocking of certain machines. How can this be modelled? By using the *Experimenter* to analyse the situation, creating several scenarios for the same variable, being this one the *Inter-Arrivaltime*.

	Variable
Variable 1	MODEL:/RawMaterials> variables/interarrivaltime

Figure 49

As the performance measure, it will be chosen the number of bread buns produced since the total production is probably the best indicator of how well a certain scenario is behaving. To do that, it will be chosen the option of *Statistic by individual object* in *Performance Measure*, and the *Input* for the *SinkBread*, since this is precisely the value representing the total production of buns. It can be seen that a name is given to the performance measure, *Bread buns produced*;, so that it appears in the graphs that will be created later.

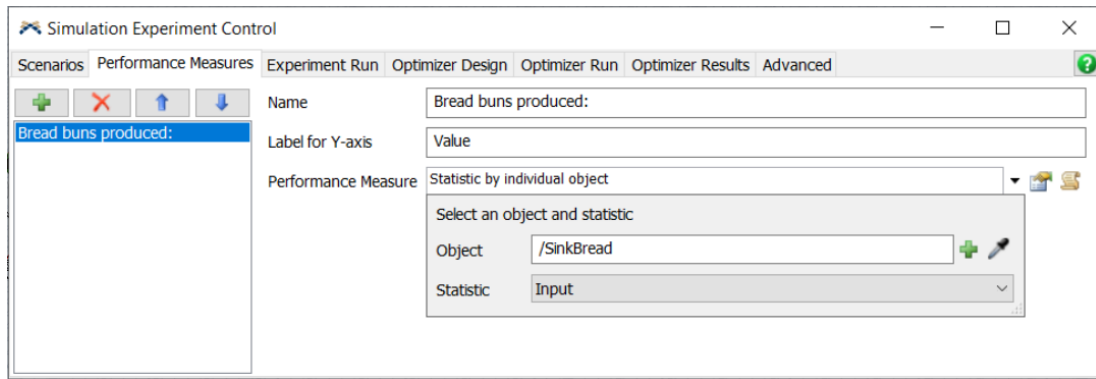


Figure 50

As the scenarios to analyse by the *Experimenter*, several values for the *Variable 1*, which represents the *Inter-Arrivaltime* of the raw materials.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Variable 1	2000	2500	3000	3500	4000	4500

Figure 51

It has been decided to perform the simulation for 3 days and then obtain the average value for the daily production. The results for the simulation of 1 day only might not be representative, since some effects, like the change from the night shift to the morning shift are not considered by the simulation. This is why 3 days has been chosen as a valid simulation time, where all effects will be taken into account.

So, to establish the conditions for the run the tab *Experimenter Run* must be accessed. These conditions are basically the *Run Time*, the 3 days expressed in seconds, and the number of *Replications per Scenario*, which will be equal to 1 since the system is deterministic and the same results would be obtained for any replication.

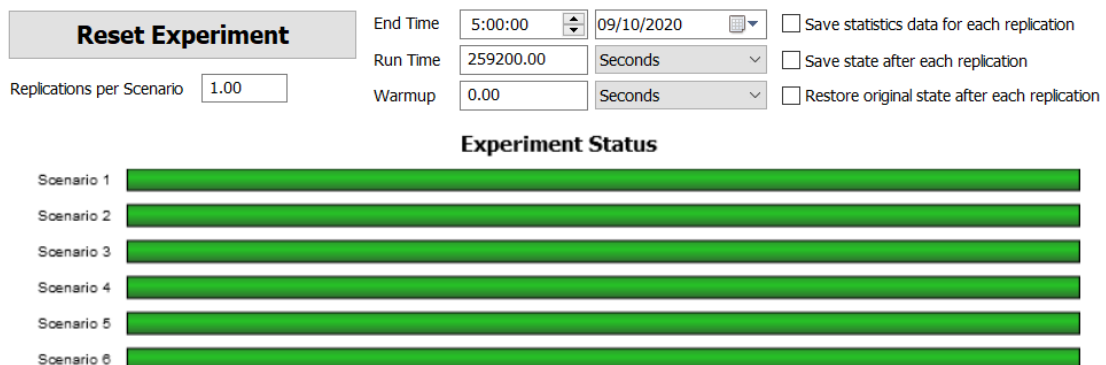


Figure 52

The fact that the bars are completely green implies that the simulation run for each of the scenarios has been finished. The *End Time* is set at 5:00:00 at the date 09/10/2020, what implies that the model has been run starting at the same time but 3 days earlier, 06/10/2020, and finishing at the *End Time*. Once the simulation of all the scenarios is complete, the results can be checked.

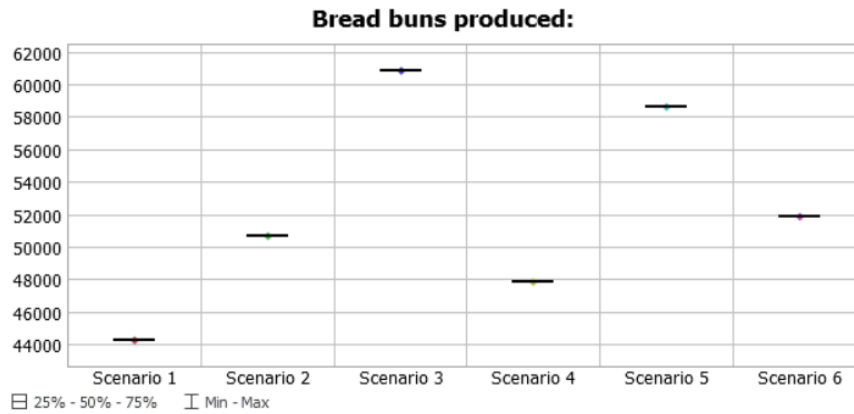


Figure 53

	Rep 1
Scenario 1	44280
Scenario 2	50640
Scenario 3	60840
Scenario 4	47880
Scenario 5	58680
Scenario 6	51840

Figure 54

It can be seen that the best behaviour, in terms of final production, is obtained with the *Scenario 3*. So, it will be assumed that the rate of arrival of the raw materials to the process is 3000 seconds, or 50 minutes. Every 50 minutes, an operator performs the task of **transport**, which represents the time it takes to carry all the materials from the warehouse to the weighing scale. It is weird to see that, for 4000 and 4500 seconds the level of production is higher than for 3500 seconds, despite expecting that because of the proximity of 3500 to 3000 the latter would yield better results than the other two. This is a clear example of how unpredictable the behaviour of the system can be regarding the variation of the *Inter-Arrivaltime* of raw materials. This evaluation is easy to be done with simulation software, but not easy in real life since the reasons for the differences in the level of production for each of the cases might not have a straightforward identification. Only experience can tell workers if they are performing at their best or not. Trusting that this experience has led them to taking the right decisions for the maximisation of the production in the real system, it will be assumed that this optimal rate has turned out to be of 1 entry of raw materials to the system every 50 minutes.

So, the total production of bread buns in 3 days is equal to 60840 bread buns. That is, an average daily production of **20280** bread buns per day.

Then, the monthly production would correspond to the daily production multiplied by the number of days of work that compose a month, which was computed above as the total number of hours worked in a month divided by the total number of hours worked in a day.

$$\text{Monthly production} = 20280 \frac{\text{buns}}{\text{day}} \cdot 25.679 \frac{\text{days}}{\text{month}} = \mathbf{520770.37 \text{ buns/month}}$$

The yearly production,

$$\text{Yearly production} = 520770.37 \frac{\text{buns}}{\text{month}} \cdot 12 \frac{\text{months}}{\text{year}} = \mathbf{6249244.44 \text{ buns/year}}$$

There is one consideration left: the real company being modelled does not only produce the bread buns that have been analysed in this study. It is true that this product is the one representing the biggest proportion of their total production, but the bakery produces other varieties of bread. That is, the total time devoted for production has not been devoted entirely to the production of this variety of bread, named *popular* according to [25]. Based on the data that this source provides, the 71.06% of the total yearly production [25, tab. 2.3] corresponds to this variety of bread. Assuming that the times for producing the different varieties of bread do not differ very much from each other, it can be considered that to produce the 71.06% of the total production it has been used around the same 71.06% of the total production time. That is, to know how many bread buns have been produced in a year for the variety that has been studied, which is *popular* bread, it can be calculated as

$$\text{Popular bread production} = 6249244.44 \frac{\text{buns}}{\text{year}} \cdot 0.7106 = \mathbf{4440713.1 \text{ buns/year}}$$

Also in [25, tab. 2.3], it can be seen that the yearly production in 2015 of this variety of bread is equal to 4459490 buns of *popular* bread. It is clear that the difference between the real datum and the obtained result is very little, but it can be computed how big this difference is by means of a percentage.

$$\% \text{ of difference} = \frac{\text{True prod. (datum)} - \text{simulated prod. (result)}}{\text{True prod. (datum)}} \cdot 100$$

$$\% \text{ of difference} = \frac{4459490 - 4440713.1}{4459490} \cdot 100 = \mathbf{0.421\%}$$

Indeed, the difference is completely negligible, and the model is absolutely succeeding in providing an accurate solution for the yearly production of the real system. The evaluation of the deterministic model for the real system is, therefore, successful. It is time now for the stochastic model.

4.3.2 Stochastic model

As it has already been discussed, the only difference between the deterministic and the stochastic models has to do with the definition of the time that is being assigned to each of the activities. It is clear that reality is stochastic, since it is virtually impossible to observe two or more repetitions of one activity having the same exact duration. It can happen, but it is just chance. This applies to activities where human workers are involved, in which this fluctuation in completely expectable, but also to machine times: a machine having a process time of 10 minutes does not imply that every process will take exactly this value. Of course, the numbers will be close to these 10 minutes, but they will oscillate in a specific range around the mean of 10 minutes (for example, some process times for this case could be 9.95 min, 10.03 min, or 10.1 min).

So, for the activities that require the use of a human operator, the standard time that was used for the definition of the process times was just a mean value. Actually, in [25, tab. AVII.1, AVII.2,

AVII.3 and AVII.4], this standard time that has been defined for the different tasks was the result of a Methods-Time Measurement (MTM) performed to those activities of interest. That is, a certain number of measurements were taken for each of the activities, according to the method being used, and the standard time obtained was just the result of applying the performance measures of the workers being tested together with the allowances for fatigue and other effects to the average value of these measurements.

According to [32], the standard time denotes the time it takes an average skilled worker, at a normal pace, to complete a task following a prescribed method. This is why the performance measure for each of the workers in every activity is required. It is basically an evaluation of how well the workers perform their assigned tasks and, for this case, it is represented by a percentage. If an operator performs fine, providing results that are considered as correct, the evaluation will be of a 100%. However, if the operator either overperforms or underperforms, the percentage will vary from the 100%. An operator with a 125% evaluation is performing better than expected. Therefore, the time it takes to perform a determined activity must be multiplied by this 1.25 so that the value of the time for the activity can be considered for the definition of the standard time. On the other hand, an operator with a 75% evaluation performs worse than expected, so it is needed to multiply its time by 0.75 [25, tab. AV.1]. Also, a value of 13% has been defined in [25, tab. AVI.1] for the total allowance to be considered. This way, it is necessary to multiply the obtained value for the mean already considering the performance measures by 1.13. This is how the mechanism for determining the standard time of an activity works, according to the method being used for this case.

However, it could be considered that the complete list of measurements for the different activities constitute a representative group to approximate the overall behaviour for the process time of the corresponding activity. Thus, if the performance measures of the workers and allowances were applied to the individual times measured instead of only to the mean, the result would be a group of standard times whose mean, of course, would still be the standard time defined for the particular process.

So, based on the tables of measurements with the applied factors to consider performance measure and allowances from [25], new tables with representative “standard times” for the activities are obtained. For the stochastic case it will not be enough to simply consider the average value of these tables, it will be interesting to be able to treat the values conforming these tables as if they belonged to a determined statistical distribution being applied to the process time of the specific activity. That is, it is required to approximate the obtained values by an existing statistical distribution, whose parameters can be modified to represent as accurately as possible the tendency of the values forming the group which is wanted to be approximated. This task is known as **probability distribution fitting** [33] and *FlexSim* includes amongst its tools a powerful distribution fitting software, named *ExpertFit*. It is accessed through the *Statistics* tab, just like the *Experimenter*.

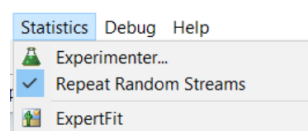


Figure 55

By clicking the *ExpertFit* option a new window opens where a *Project 1* has already been created by default and it is ready for new *Project Elements* to be introduced within. As mentioned, the desired *Analysis Type* for these elements will be the *Fit distributions to data* option. By clicking *OK*, the *Project Element* is defined as a *Data Analysis*.

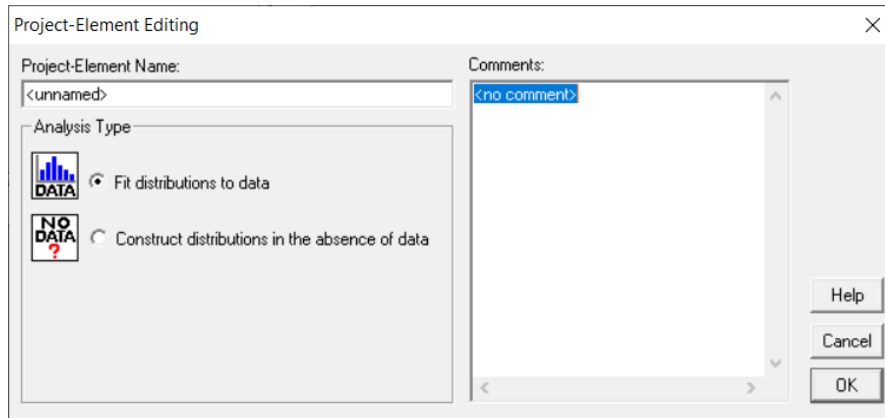


Figure 56

This *Data Analysis* is opened in order to introduce the set of data which is wanted to be approximated. The data can be read from a text file (*Read Data from File*), either in .dat or .txt format. By clicking *Apply*, any already created file with these characteristics can be introduced to the analysis. Therefore, to approximate each of the activities' process time by a distribution, it will be necessary to create a text file with all the values for each of the cases. For this study, .txt format has been used.

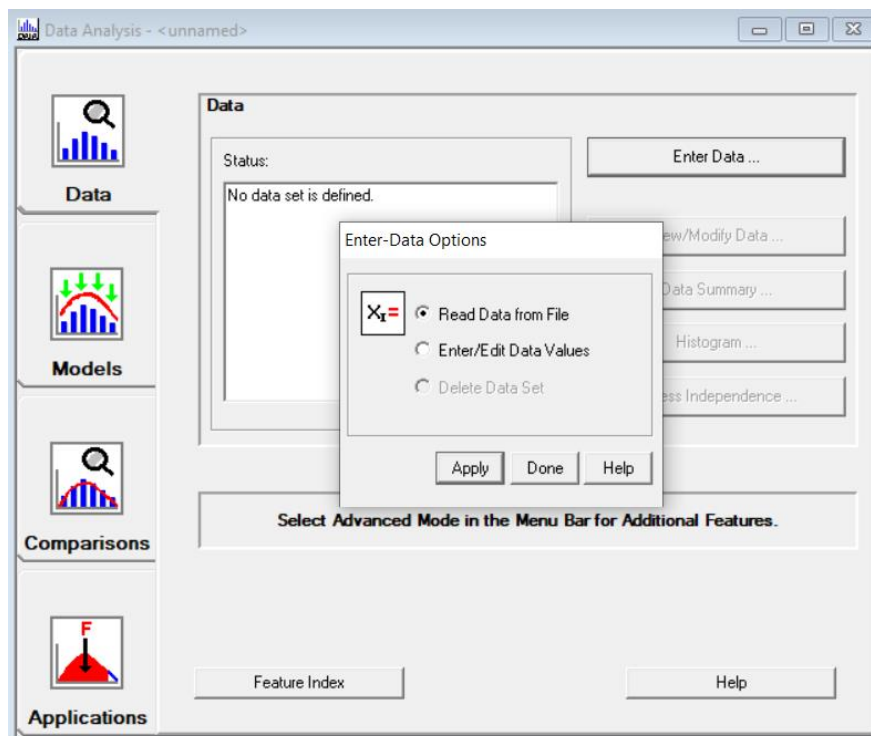


Figure 57

As an example, it is going to be developed the case for the process time of the first activity whose time has been measured in the production system: the **transport**. Right after choosing the text file with all the values corresponding to the “standard time” data for the transport, a table appears showing a summary of the introduced data.

Data-Summary Table

Data Characteristic	Value
Source file	Transport
Observation type	Real valued
Number of observations	31
Minimum observation	509.51700
Maximum observation	662.27040
Mean	587.19284
Median	584.26650
Variance	1.661.74505
Coefficient of variation	0.06942
Skewness	0.07937

Copy
Print
Help
Done

Figure 58

Some relevant information is provided by this *Data-Summary Table*, like the mean value or the variance of the sample.

The next step is to click, in *Models*, the option that says *Automated Fitting...* By doing that, the following results are obtained.

Automated-Fitting Results

Model	Relative Score	Parameters
1 - Log-Logistic(E)	84.68	Location 223.00143
		Scale 362.37007
		Shape 15.78790
2 - Log-Logistic	83.87	Location 0.00000
		Scale 585.73426
		Shape 25.53588
3 - Lognormal	75.81	Location 0.00000
		Scale 585.82150
		Shape 0.06846

32 models are defined with scores between 1.61 and 84.68

Absolute Evaluation of Model 1 - Log-Logistic(E)
 Evaluation: Good
 Suggestion: Additional evaluations using Comparisons Tab might be informative.
 See Help for more information.

Additional Information about Model 1 - Log-Logistic(E)
 "Error" in the model mean relative to the sample mean -0.58116 = 0.10%

Copy
Print
Help
Done

Figure 59

Here, the distributions that fit the set of values introduced for the analysis are listed, and they are ordered according to their *Relative Score*, which is determining how well the different distributions adjust to the data. The best three candidates have been introduced. In this case, the best of them, *Model 1*, is represented by a *Log-Logistic(E)* distribution whose *Evaluation* is *Good* and the “*Error*” in the model mean relative to the sample mean has a value of 0.1%.

Now, in *Comparisons*, it is possible to compare the data sample with the chosen model that corresponds to the best fitting.

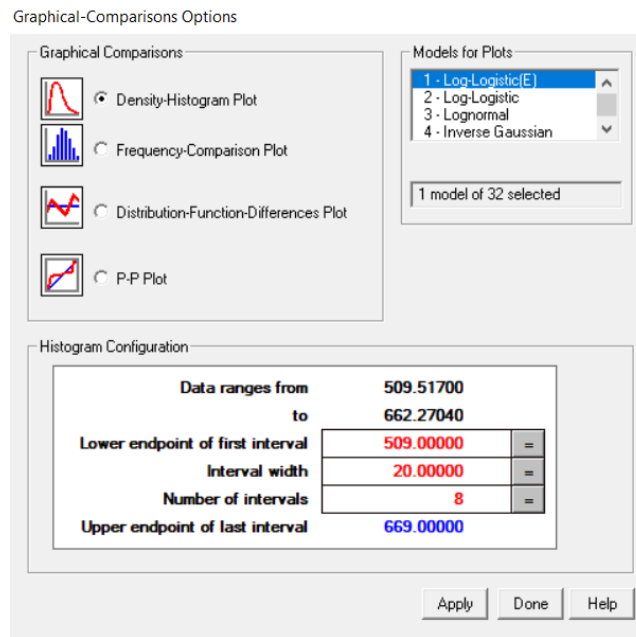


Figure 60

By applying the previous, the next *Density-Histogram Plot* is attained for the *Graphical Comparison*.

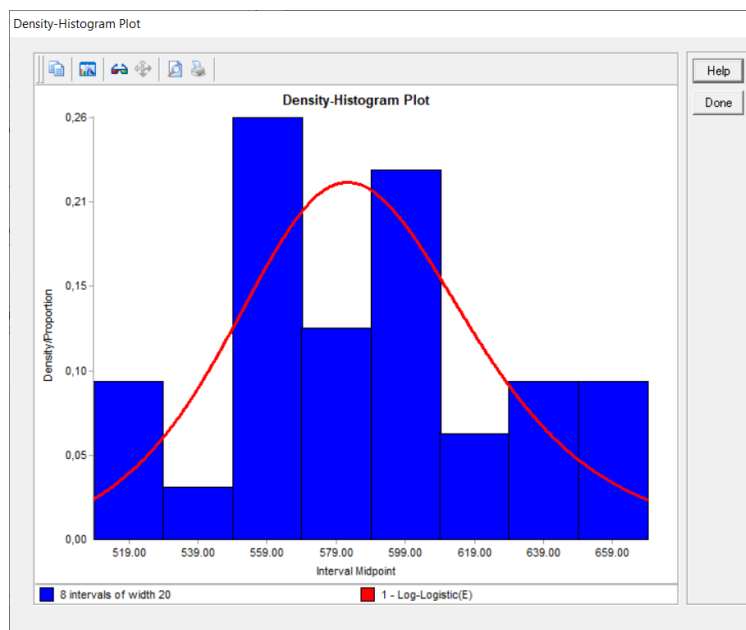


Figure 61

Another option to evaluate how well the fitting of the distribution to the data is might be the *Goodness-of-Fit Tests*. Some well-known tests are included in this option of *ExpertFit*.

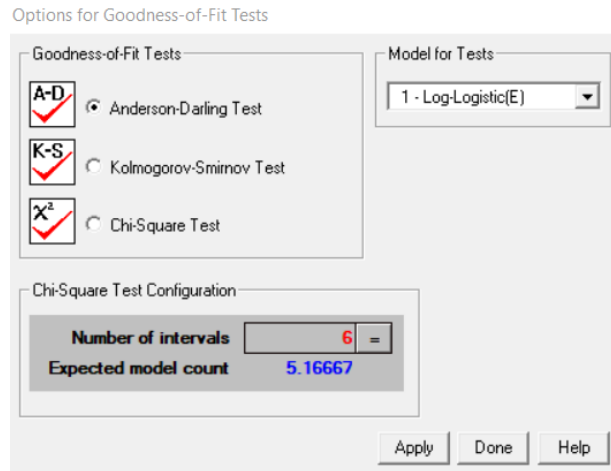


Figure 62

If the selected *Anderson-Darling Test* is applied to *Model 1*, the following results are obtained.

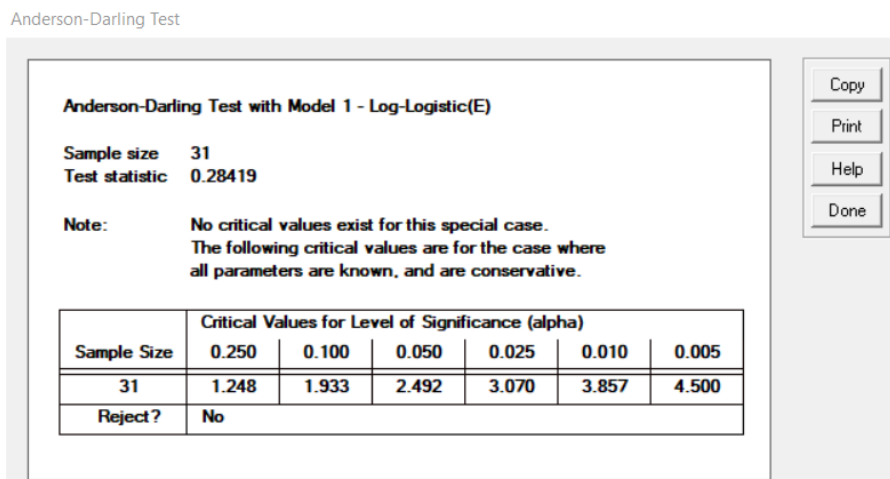


Figure 63

This test provides a series of results that end up answering the question *Reject?* with a *No*. This means that the test proves the fitting is good, otherwise it would have chosen to reject the fitting instead of not rejecting it. The same answer is given by the other two tests. Therefore, the fitting is definitely good enough.

The final task would be to extract from this analysis the appropriate expression to represent the process time of the corresponding activity so that it follows the characteristics of the statistical distribution that has been chosen as the best approximation. To do that, in *Applications*, in the box for *Use a Specified Distribution (Model)* in must be clicked the button that says *Simulation Representation...* Then, a new window will open where the *Model to Represent* will already be chosen by default and it can be chosen which Simulation Software is wanted for the obtention of the representation of the model. The only one available is the one of interest, which is precisely *FlexSim*.

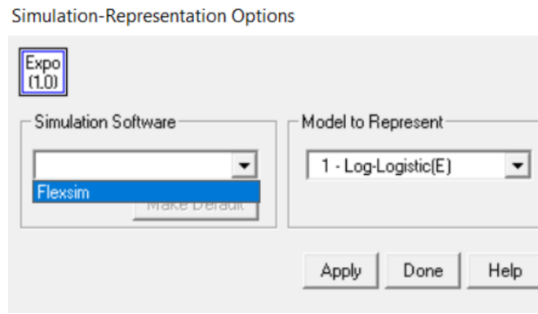


Figure 64

By applying the representation, the following is achieved.

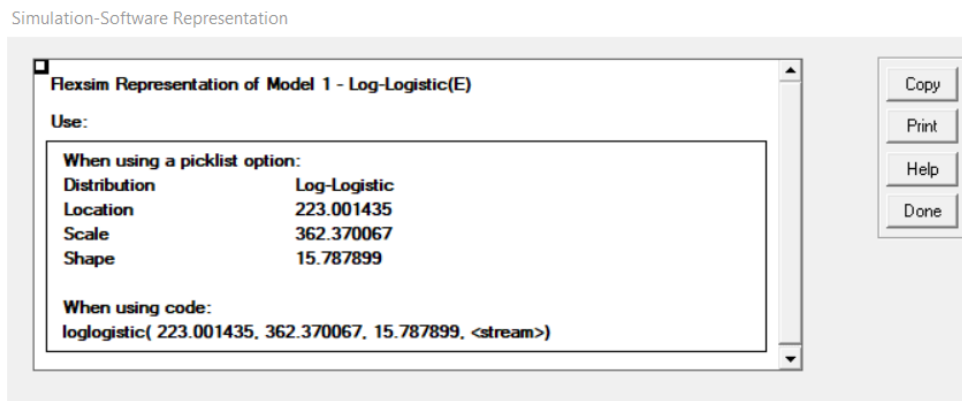


Figure 65

Then, as shown by the previous table, in order to assign a *Process Time* to a determined activity, in this case the transport, it can be done by means of a *picklist option* or *using code*. The first option would look like this in *FlexSim*.

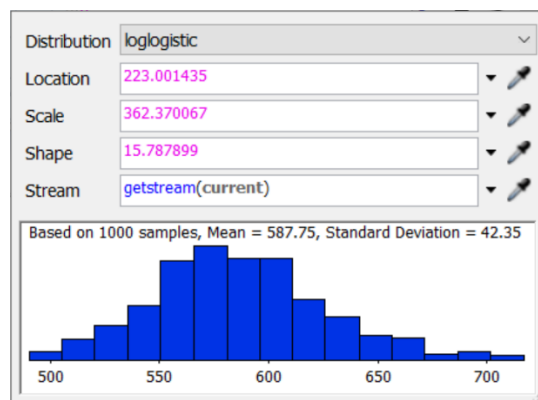


Figure 66

The second one would look like this.

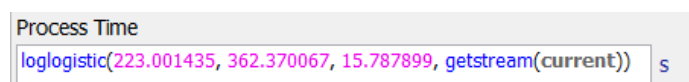


Figure 67

The process for the assignation of a statistical distribution to appropriately model the Process Time of the transport is finished. This same procedure must be applied to each of the activities that have undergone the Methods-Time Measurement. Depending on the data for the different cases, different distributions might be obtained for the representation of each of them.

Apart from the activities involved in the MTM, there are left to specify the machine times, now that it has been understood that the mean value cannot happen repeatedly, and the *Process Time* for the activities that, despite being performed by human operators, have not been part of the MTM study.

So, for the machine times the following has been considered. The kneading takes an average duration of 15 minutes, and so do the baking and cooling stages. In order to consider the variations from this mean value in the real life, it has been chosen to describe the behaviour of the *Process Time* for this activity with a **normal distribution**. A variation of plus and minus 30 seconds with respect to the mean has been considered as reasonable so, knowing that these limits represent the value $\mu - 3\sigma$ and $\mu + 3\sigma$, μ being the mean of the distribution and σ being the standard deviation, the value for the latter must be of 10 seconds (with the mean value kept at 900 seconds). This is how the *picklist option* for the representation of this distribution looks like in *FlexSim*.

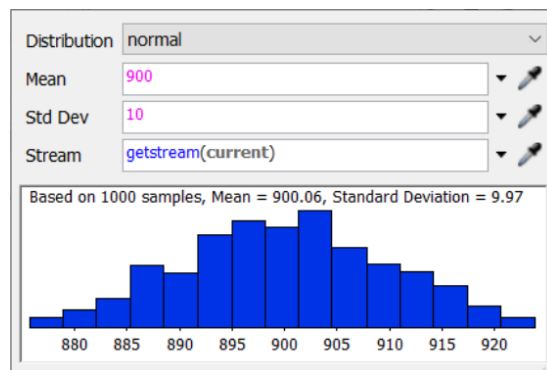


Figure 68

Something similar has been assumed for the fermentation stage, where a normal distribution has been defined with a mean value of 1800 seconds and a standard deviation of 20 seconds, what allows for a variation from the average 30 minutes of plus and minus 1 minute. This ends with the machine times. Now, the extraction of the dough and the packing stages are the only ones remaining.

For the two parts of the packing, normal distributions have been chosen with a mean of 30 seconds and a standard deviation of 2 for the first part, and a mean of 5 seconds and a standard deviation of 0.33 seconds for the second. There is no data available for this last step, so these values have been picked since they seem reasonable and appropriate for the activities that they are modelling.

Finally, for the extraction of the dough a different approach has been taken. Despite not being analysed in the MTM study, there were available two measurements of this activity [25, tab. AVII.1 and AVII.3]: 90 seconds and 82 seconds. For the deterministic model, the average value of this two, which is 86 seconds, was used. However, for the case of the stochastic model, it has

been decided that an appropriate distribution for the representation of the *Process Time* of this activity could be a uniform distribution with a minimum value of 82 seconds and a maximum of 90 seconds. With this, it has been completed the definition of the *Process Time* for all the activities in the stochastic model. Then, it is time to proceed with the simulation.

By running the simulation for the stochastic case with the exact same characteristics defined for the *Experimenter* that was used in the deterministic case, it is observed that the tendency of the results as the variable being modified for the different scenarios, which is the *Inter-Arrivaltime* of the raw materials, increases is still going up for the last case being analysed in the *Experimenter*. It seems reasonable to modify some of the scenarios so that this tendency can be appropriately studied. So, the first 2 scenarios will be erased and 3 more will be added to the right of 4500 seconds.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7
Variable 1	3000	3500	4000	4500	5000	5500	6000

Figure 69

The result that was obtained as the best for the deterministic case was the one for the scenario of 3000 seconds. However, it might occur that, for the stochastic case, this scenario is no longer the most favourable. In that case, the reason for the change would be the variability that has been applied to the *Process Time* of all the different stages of the system.

As it was mentioned for the previous analysis, the use of the *Experimenter* is particularly useful for the stochastic case. This usefulness has to do with its capability of creating several *Replications per Scenario*. For the deterministic case, only 1 replication per scenario was created, since no variation of the results was expected because of all the data and variables defining the model being deterministic. However, now that statistical distributions have been applied to the processes to model their *Process Time*, the creation of several *Replications per Scenario* becomes mandatory.

The *Experimenter* is able to apply randomness to each of the replications that are created for the scenarios. That is, each replication will be considering, for each of the stages, a specific *Process Time* which will have been chosen randomly from all the possibilities that are observed by the statistical distribution that applies for each determined activity. So, unlike the case for the deterministic model, each replication will yield a different result and all of them must be considered as possible and valid results, since they are all obtained as a consequence of a random combination of the times of the activities in the system.

Therefore, for the evaluation of the stochastic model with the defined scenarios, 3 days will be simulated and, the main difference with the simulation of the deterministic model will be the choice of 50 *Replications per Scenario*, in order to obtain a representative sample of results so that the subsequent analysis is meaningful.

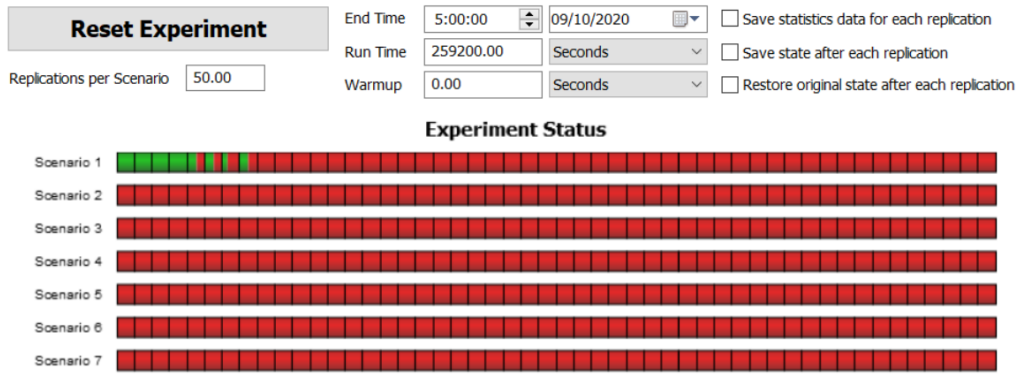


Figure 70

It is observed how each of the bars representing the seven scenarios is divided in 50 smaller parts, which denote the *Replications per Scenario*. Unlike the case of *Figure 50*, where the experiment of the six defined scenarios for the deterministic model was already finished, the experiment for this case is still ongoing, what can be noticed by observing that most of the bars is still in red colour. As a matter of fact, only the first 4 replications for *Scenario 1* have been finished, and the second 4 replications are now being developed, what is represented by the evolution of the green colour within the bars. Of course, the fact that 350 replications are required to be computed, instead of the 6 scenarios with 1 replication each of the previous case, implies that the computational cost is significantly higher and, therefore, the time it takes to perform the complete simulation is considerably longer.

Once the simulation is finished, the following results are obtained.

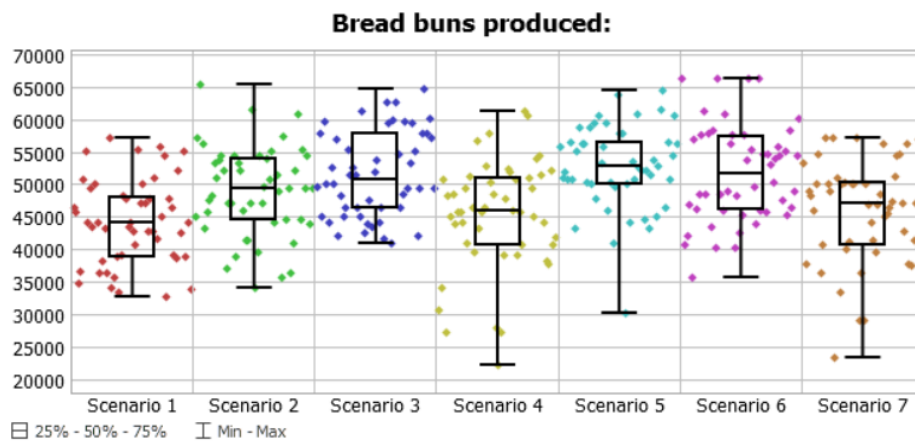


Figure 71

Bread buns produced:

	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	42729	< 44288	< 45846	6545	32760	57240
Scenario 2	47253	< 48890	< 50526	6870	34200	65320
Scenario 3	49992	< 51562	< 53132	6593	41040	64800
Scenario 4	43344	< 45356	< 47368	8448	22320	61200
Scenario 5	51207	< 52734	< 54260	6411	30240	64440
Scenario 6	50013	< 51792	< 53571	7469	35640	66240
Scenario 7	44057	< 45929	< 47801	7861	23400	57240

Figure 72

It can be highlighted that there are 3 scenarios that are giving the best results, according to the *Replications Plot*. These scenarios are number 3, number 5 and number 6. All their mean values are above 50000, with *Scenario 5* having the higher *Mean* with a value of 52734 (given within a 90% confidence interval). However, the biggest maximum value that is observed for all the scenarios happens in *Scenario 6*, with a value of 66240, and the smaller one for the 3 scenarios providing the best solutions occurs in *Scenario 5*, with a value of 30240. The *Sample Std Dev*, referring to the sample standard deviation, describes how the variation of the results within the scenarios is. Since this value depends on the mean, the actual variability of the results will need to be measured by means of the *Coefficient of Variation (CV)*, computed by the quotient of the standard deviation over the mean. The smallest fluctuation, represented by the smallest value for the CV, is observed for *Scenario 5*. Hence, based on the fact that *Scenario 5* has the biggest *Mean* with the smallest CV, it can be concluded that it is the scenario providing the best results for the total production. Then, it can be said that the best option is for the operators to introduce a batch of raw materials to the production line every 5000 seconds (or 1 hour, 23 minutes and 20 seconds).

Knowing that 5000 seconds has provided the best solution, it can be further analysed if the best situation happens at 5000 seconds or somewhere around that value. For that, 9 scenarios equally spaced between 4500 and 5500, both included, have been studied, yielding the following.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8	Scenario 9
Variable 1	4500	4625	4750	4875	5000	5125	5250	5375	5500

Figure 73

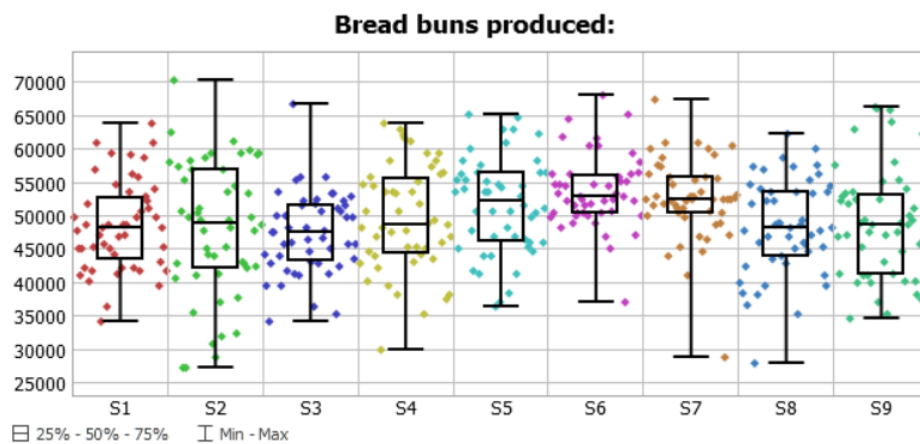


Figure 74

Bread buns produced:						
	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	46910	< 48501	< 50092	6680	34200	63720
Scenario 2	45616	< 48020	< 50425	10097	27360	70200
Scenario 3	45974	< 47400	< 48826	5988	34200	66600
Scenario 4	47763	< 49639	< 51515	7876	29880	63720
Scenario 5	49864	< 51576	< 53287	7186	36360	65160
Scenario 6	52369	< 53650	< 54930	5378	37080	68040
Scenario 7	51042	< 52472	< 53901	6003	28800	67320
Scenario 8	46613	< 48309	< 50006	7124	28080	62280
Scenario 9	46883	< 48849	< 50815	8254	34560	66240

Figure 75

Now, the highest *Mean* and smallest CV is observed for *Scenario 6*, which corresponds to an *Inter-Arrivaltime* of the materials to the production line of 5125 seconds (or 1 hour, 25 minutes and 25 seconds).

Some scenarios of this second evaluation, despite being the same as some of the scenarios evaluated in the previous one, do not provide the same results in both cases. *S1*, *S5* and *S9* in the second, which correspond to scenarios 4, 5 and 6 in the first, respectively, present slight changes in the second evaluation with respect to the first. This is a clear example of how the randomness is applied to the different processes by the *Experimenter*. The results must be relatively close, which is the case, but they must not coincide in any case. Otherwise, the random generation of replications performed by the *Experimenter* would not be working appropriately.

As a conclusion, it can be stated that the stochastic model is providing a less optimistic result than the deterministic model did. If the mean value of 53650 obtained as the best case for this second evaluation is considered, if compared to the 60840 obtained for the deterministic model, a drop of nearly 12% is observed. However, the value of the deterministic model would still be regarded as possible by the stochastic model since, for the chosen case, the maximum value obtained for the sample was 68040.

If instead of comparing the best solution for the deterministic case with the best one for the stochastic case it is compared to its corresponding scenario, the one dealing with the same *Inter-Arrivaltime* of 3000 seconds, it can be observed that the result for the deterministic model is not even considered within the scope of the solutions obtained for the stochastic one. It is clear then that the first solution was, for any reason, too optimistic, and it was not representative of the real overall behaviour of the system.

Moreover, despite presenting this considerable difference in terms of the mean value for the best cases in both simulations, the stochastic model represents a realistic model, as it has already been discussed, since it is taken into account any possible time fluctuation within the processes conforming the complete system. The deterministic model, despite apparently representing the real system so accurately, as it was suggested by the proximity of the yearly production simulated to the real data, is not realistic whatsoever. Again, it has served its purpose of providing a “non-realistic” simulation of the real system, where only average times were used for the processes but, most importantly, this deterministic model has served as an intermediate step which has led the way for the attainment of the final model, this one being the stochastic model whose definition has just finished.

To sum up, this model can be considered as appropriate and, therefore, it will be available for the study of **What if?** scenarios that could be applied to the existing production system.

5. ANALYSIS OF TEST SCENARIOS WITH THE SIMULATION MODEL

5.1 TEST SCENARIO NUMBER 1

“What if the current working schedule was modified in order to guarantee that all the 10 operators do not exceed the 8 hours of work per day?”

The current working schedule being followed in the production system is

Shift	Schedule	Number of workers
Morning	5:00-14:00	1
Afternoon	14:00-22:00	5
Night	21:00-5:00	4

Table 6 [25, tab. 2.1]

The morning shift is composed of 9 hours, whereas the other two shifts are already composed of 8 hours. The new proposed schedule must achieve the reduction of the morning shift from 9 to 8 hours so that all the three shifts last 8 hours.

Apart from the fact that the current morning shift lasts 9 hours, it can be noted that the afternoon and night shifts interfere for 1 hour, between 21:00 and 22:00. This explains why the 9-hour shift was possible in the first place, since the sum of the other two shifts, despite lasting 8 hours each, covers 15 hours instead of 16. Then, the shortening of the morning shift will require the afternoon shift to be moved forward by 1 hour and, as a result, no interference will remain between this shift and the night one.

The last consideration has been the modification of the number of workers present within each of the shifts. Searching for a more balanced distribution, 3 workers will work in the morning shift, 4 workers in the afternoon shift, and the remaining 3 workers in the night shift. The modification of Table 6 [25, tab. 2.1] to represent the new proposed schedule will be as follows.

Shift	Schedule	Number of workers
Morning	5:00-13:00	3
Afternoon	13:00-21:00	4
Night	21:00-5:00	3

Table 7

The choice of the afternoon shift as the one having 4 operators instead of 3 has been arbitrary. In fact, it is expectable that, no matter which of the three shifts is composed of 4 operators, the results will be similar.

Once the characteristics for the first test scenario have been defined, it is a matter of modifying the already built final model of the real system in order to meet the requirements of this scenario. In this case, it is necessary to change the parameters that were assigned to the *Time*

Tables, the FlexSim tool being responsible for the correct application of the working schedule to the process.

As it was seen in the definition of the model, the appropriate working schedule was achieved by determining the resting times for the different groups of workers. Accordingly, the resting times for this new case must be from 13:00:00 to 5:00:00 in the morning shift, from 5:00:00 to 13:00:00 and from 21:00:00 to 5:00:00 in the afternoon shift, and from 5:00:00 to 21:00:00 in the night shift. These changes can be easily applied by shortening and translating the times defined for the real case model.

After rearranging the shifts' duration, it is time to modify the groups of workers to which the Time Tables are applied. Here is how the new organisation will look like: Time Table1 will concern the operators assigned to the morning shift, Time Table2 and Time Table3 (which is not shown in the figure) will be applied for the afternoon shift, and TimeTable4 will refer to the night shift.

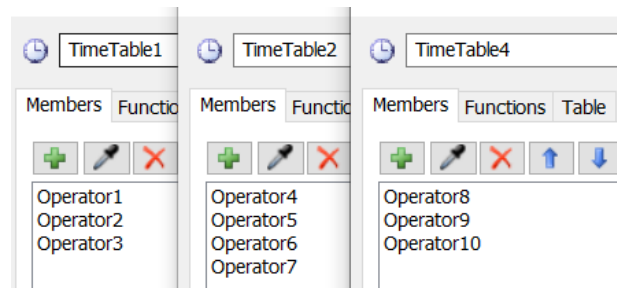


Figure 76

The test scenario is ready to be run. The exact same situation will be analysed within the Experimenter for this test scenario as the one that was studied for the real model. Hence, a direct comparative analysis of the results will be possible.

The simulation of the test scenario number 1 provides the following results.

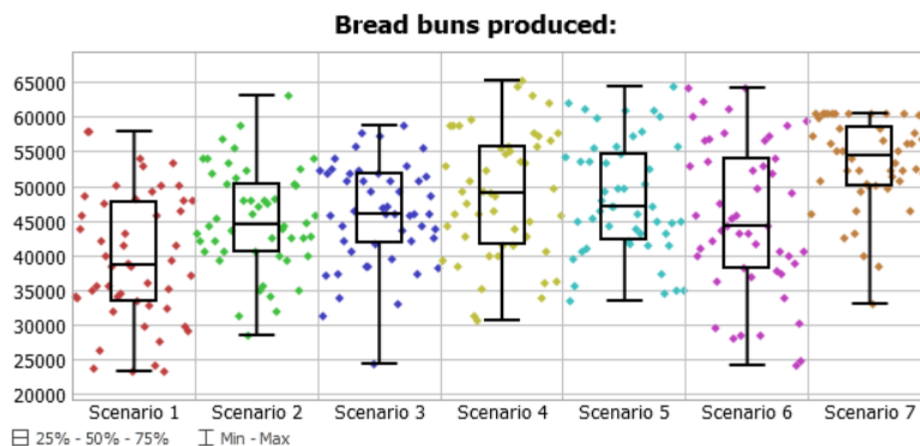


Figure 77

Bread buns produced:

	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	37396	< 39608	< 41819	9286	23400	57960
Scenario 2	43485	< 45238	< 46990	7358	28440	63000
Scenario 3	44191	< 45936	< 47681	7327	24480	58680
Scenario 4	46668	< 48800	< 50933	8955	30600	65160
Scenario 5	46311	< 48260	< 50210	8185	33480	64440
Scenario 6	42876	< 45459	< 48042	10847	24120	64080
Scenario 7	51751	< 53314	< 54877	6563	33120	60480

Figure 78

The best solution is provided by *Scenario 7*, since it has the biggest *Mean*, expressed with a 90% confidence interval around the value 53314, and the smallest coefficient of variation.

To start with, it is remarkable that the best solution does no longer occur around the value of 5000 seconds of *Inter-Arrivaltime* (*Scenario 5*) as it did for the case of the real system. It now occurs for 6000 seconds instead. Before proceeding with the comparative analysis of the results, the values around the *Inter-Arrivaltime* of 6000 seconds will be studied, in a similar way to how it was done for the model of the real case.

The new scenarios to be studied for the rate of arrival of the raw materials to the system are defined.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8	Scenario 9
Variable 1	5500	5625	5750	5875	6000	6125	6250	6375	6500

Figure 79

The new results provided by the *Experimenter* look like this.

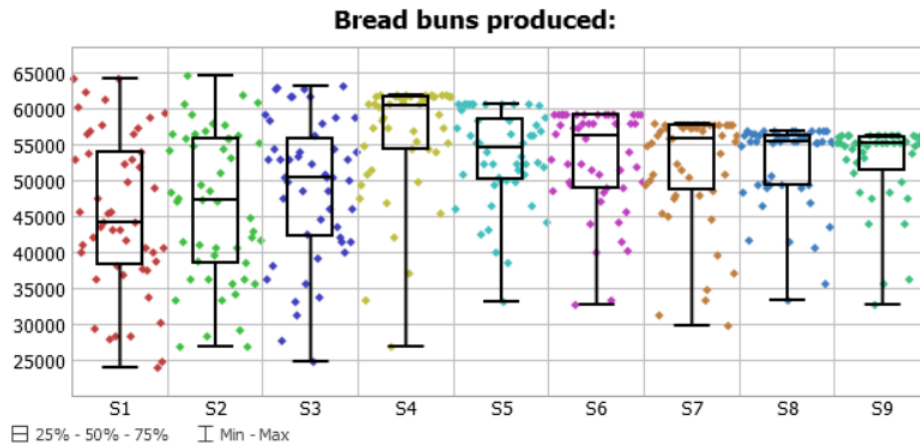


Figure 80

Bread buns produced:						
	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	42876	< 45459	< 48042	10847	24120	64080
Scenario 2	43936	< 46378	< 48821	10256	27000	64440
Scenario 3	46729	< 49058	< 51388	9780	24840	63000
Scenario 4	54296	< 56192	< 58087	7959	27000	61920
Scenario 5	51751	< 53314	< 54877	6563	33120	60480
Scenario 6	51273	< 52939	< 54605	6995	32760	59040
Scenario 7	50045	< 51876	< 53707	7690	29880	57960
Scenario 8	51036	< 52434	< 53832	5870	33480	56880
Scenario 9	51119	< 52460	< 53801	5631	32760	56160

Figure 81

The highest *Mean* is obtained for *Scenario 4*. However, unlike all the previous cases that have been analysed, this scenario does not provide the smallest CV, which is the case for *Scenario 9*. Anyway, the difference in terms of the mean comparing the one for *Scenario 4* to the rest is big enough to understand that this scenario is the one providing the most interesting solution. That is, an *Inter-Arrivaltime* of 5875 seconds (or 1 hour, 37 minutes and 55 seconds).

One important remark to make is that, unlike the case for the simulation of the model for the real system, the random streams applied to the scenarios which are being simulated again, which are the cases of 5500 and 6000 seconds of *Inter-Arrivaltime*, are being repeated. That is, no different solutions are obtained for these cases in comparison to the previous simulation since the same random points for the analysis are being selected by the *Experimenter*. Indeed, it can be observed that the obtained results for *Scenarios 1* and *5* in this second study coincide with the results for *Scenarios 6* and *7* in the first one. Similarly to what happened with the real system's model, if different random streams had been applied to the scenarios being studied twice, slightly different results would have been appreciated, but the overall behaviour would have still been pretty much the same.

So, once the best behaviour for the first test scenario has been obtained, it seems reasonable to compare it to the best solution that was obtained in the simulation of the model for the real case. The values to be compared are: **56192** for the first test scenario and **53650** for the real system's model. These values, of course, correspond to the total production in 3 days of work.

An increase of almost **5%** of the production is obtained. This is already good news, since the cost of rearranging the working schedule is almost negligible and, therefore, the application of the change would be translated into an immediate increase of the production, with its corresponding benefits to the production system.

Moreover, it must be reminded that the new working schedule is only composed of 8-hour shifts. The 9-hour morning shift has now been removed by this test scenario. Two possibilities are analysed. For the shake of simplicity and in order to properly address *Operator1* in the following paragraphs, it will be considered as a male operator.

- The **first**, the operator working on his own for 9 hours during the morning shift is actually the owner of the company, or an important associate, therefore, the remaining 9 workers are his employees. The modification of the morning shift is beneficial to him since he can reduce his working hours without jeopardising the daily production of the system. Moreover, the change does not imply that the

salaries that he must pay to the other 9 operators are increased, since this option considered the maintenance of the working shifts in 8 hours per day. On the contrary, he will be paying the same for a more interesting distribution of the working hours. The interference from 21:00 to 22:00, which presented a picture of 9 operators working at the same time, what must definitely cause a lot of inefficiencies (what is agreed by [25]), no longer exists and, overall, the new proposed schedule seems to be much more balanced and has proven to perform successfully.

- The **second**, *Operator1* is another employee. Then, it would be a matter of communicating to him that his working hours should be reduced from 9 to 8 hours and, as a consequence, his salary should also be reduced. Therefore, the employer will no longer have to pay for 81 hours of daily work, but only for 80. This daily saving of 1 hour of salary, added to the increase of production, clearly implies that the system would benefit from the application of this change.

As a conclusion, it could be said that, after analysing the results provided by the test scenario number 1, the proposed changes, in this case the new working schedule, are probably worth to be implemented.

5.2 TEST SCENARIO NUMBER 2

“What if the supplier of raw materials restricts the daily provision to an amount that only allows to produce 12 batches per day?”

So far, the restrictions imposed by the raw materials’ supply have not been considered. It has been stated throughout the whole study that the operators are free to travel to the warehouse in order to pick a new batch of raw materials according to the *Inter-Arrivaltime* defined as the best for the case being analysed. No limitations, what is in accordance with the data provided by [25], where no restrictions concerning the raw materials were specified.

Of course, this represents a perfect case. The workers keep their most prolific pace and they always have available raw materials. Ideally, the suppliers provide just as many products as the ones that can be processed by the system, therefore, no stockage is produced and there are no unproductive times where the workers are waiting for the arrival of a new batch of raw materials.

However, for the test scenario number 2 it is going to be assumed that the supplier is imposing a restriction in the supply. Daily, a shipment of raw materials will be introduced into the warehouse of the system. This shipment will allow to produce exactly 12 batches. Knowing that no more materials will be received until 24 hours have passed, it seems reasonable to establish that the entries of raw materials into the system must take place every 2 hours (or 7200 seconds). This way, the maximum 12 batches per day to process are equally distributed along the 24 hours of daily work that are available.

Therefore, the initial step for the analysis of this second test scenario will be the evaluation of the *Experimenter* for an *Inter-Arrivaltime* of 7200 seconds. This change is directly applied to the *Experimenter* of the real system’s model and run.

	Variable	Scenario 1
Variable 1	MODEL:/RawMaterials>variables/interarrivaltime	7200

Figure 82

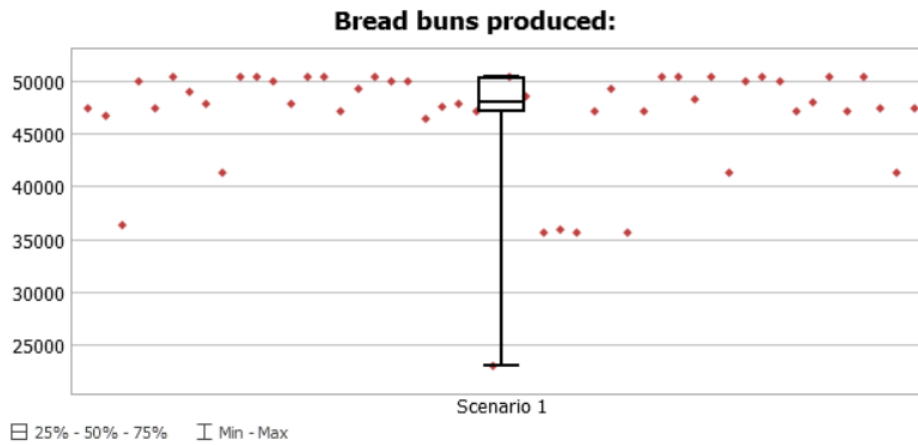


Figure 83

Bread buns produced:						
	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	45316	< 46639	< 47962	5556	23040	50400

Figure 84

As it could be expected, the production has been considerably reduced because of the restrictions imposed by the availability of raw materials. In comparison to the model for the real case, where the most interesting *Inter-Arrivaltime* turned out to be a value around the 5000 seconds, it was predictable that this big difference would translate into this important reduction of bread buns produced in 3 days. In terms of the *Mean* value, the best case for the real system’s model exhibited an average of **53650**, whereas the new situation being analysed in this test scenario number 2 exhibits a value of **46639**. That is, the new situation presents a drop of almost **14%** in terms of the total production.

It will be assumed that the situation for the current test scenario becomes a long-term reality and nothing can be done to recover the “ideal” working state that was achieved for the model of the real system.

Knowing how the actual system behaves, something that has been possible thanks to the study that was performed with the model for the real system, it is clear that the direct solution to the current “problem” with the availability of raw materials would be the acquisition of more of these same products. It could be achieved by contacting new suppliers and, if necessary, by working with different suppliers at the same time.

Nevertheless, for the intended study of this test scenario number 2 the only possibility will be to agree to this reduction imposed by the current supplier and no other suppliers will be available to work with. This situation could be a coarse representation of how an economic crisis could affect the system, in this case, by striking directly to the sector in charge of the delivery of raw materials.

Then, a different approach should be taken for the current situation. The 12 batches per day limitation has become the new reality, at least in the medium/long-term. It has been observed that the difference in terms of the *Inter-Arrivaltime* if the real case is compared to the current case is pretty big. Apart from the prediction of a reduction in total production it can be understood that lots of unproductive times might have arisen within the working shifts. Then, the following question could be asked: could the system keep up with the new production pace even though some operators were fired?

Normally, firing workers is a very drastic measure but it can be justified in the event of exceptional situations like the one being analysed. In order to study the new possible cases where some operators have been fired, a new tool will be used in *FlexSim: Groups*.

Basically, a *Group* will be created for each of the three working shifts. Each one will be composed of just 1 operator, so for the analysis of the current test scenario, all the remaining operators can be erased from the model. The following picture shows how a *Group* includes an operator. The same occurring for *Group1* and *Operator1* (morning shift) will be happening to *Group2* and *Operator2* (afternoon shift), and *Group3* and *Operator7* (night shift). As mentioned, operators 3, 4, 5, 6, 8, 9 and 10 have been erased.

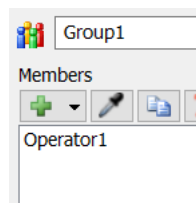


Figure 85

The great value of using *Groups* is provided by the possibility in the definition of the variables within the *Experimenter* of specifying the *Number of Objects in Group* for each of the scenarios that are wanted to be studied. Therefore, simply by typing a number in the empty cells to the right of the defined variables it is implied that, for each scenario, the process will use as many operators of the same kind as the one included inside the *Group* being selected as the typed number reflects. Therefore, the following scenarios will be considered.

	Variable	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Variable 1	MODEL:/RawMaterials>variables/interarrivaltime	7200	7200	7200	7200	7200	7200
Variable 2	Nr Objects in Group1	1	1	1	1	1	1
Variable 3	Nr Objects in Group2	4	5	4	5	4	3
Variable 4	Nr Objects in Group3	4	3	3	2	2	3

Figure 86

It is reminded that the real system presents 1 worker in the morning shift, 5 workers in the afternoon shift and 4 workers in the night shift. As a matter of common sense, the reduction of the number of operators in *Group1* has not even been considered since it would have implied the total disappearance of the production in the morning shift. Then, different options of dismissals have been proposed by the defined scenarios: the first two scenarios consider 1 dismissal; the next two, 2 dismissals; and the final two, 3 dismissals.

The following results are obtained.

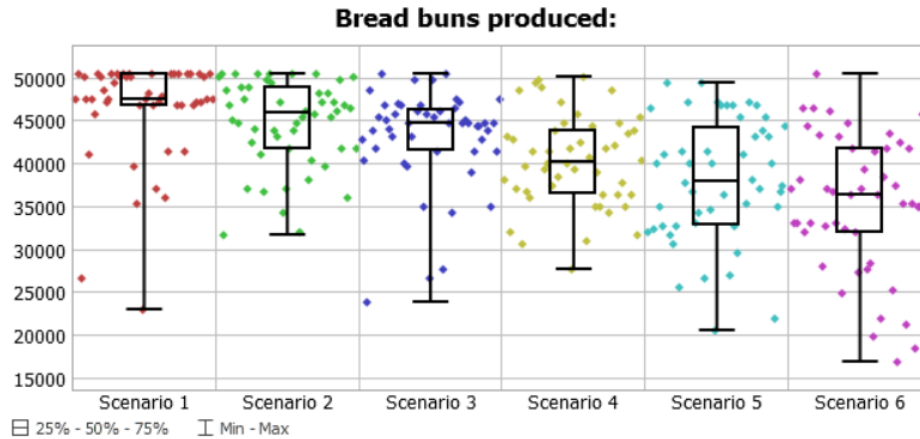


Figure 87

Bread buns produced:						
	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	45035	< 46444	< 47853	5914	23040	50400
Scenario 2	43442	< 44657	< 45872	5101	31680	50400
Scenario 3	41777	< 43123	< 44470	5654	23760	50400
Scenario 4	39038	< 40320	< 41602	5385	27720	50040
Scenario 5	36428	< 38146	< 39863	7212	20520	49320
Scenario 6	33456	< 35405	< 37354	8185	16920	50400

Figure 88

It is observed that *Scenario 1* represents the best option if 1 dismissal is considered, *Scenario 3* for the case of 2 dismissals considered and *Scenario 5* for the case of 3 dismissals.

Now that the results have been obtained it becomes a matter of analysing each of the cases: is the company willing to sacrifice even more production, what will definitely happen if operators are made redundant (it is indeed observed in the obtained results), in exchange for the considerable saving that would be obtained by having to pay less salaries?

Often, when this type of dilemma arises, companies take the course of action of analysing their **productivity** indicators [34]. There are lots of different indicators that can be obtained, but the main idea of most of them is the direct comparison of the earnings being obtained by the company with the costs that are required to obtain the total production that is generating income: global productivity. Normally, it is represented by the quotient of the sales revenue over the costs.

It seems reasonable to believe that the dismissal of 1 operator is providing an interesting result (*Scenario 1*), since the reduction in terms of the production is almost negligible in comparison to the case of 10 operators working under the imposed restrictions. An average production of 46639 goes down to an average 46444 (a drop of less than 0.5%) and 1 full 8-hour salary would be saved. The computation of the corresponding productivity indicators, in this case, the workforce productivity, would probably recommend the dismissal of 1 operator from the afternoon shift since the fact that the income is kept almost constant and the costs are reduced implies that the productivity would increase. The same way the global productivity represents the sales revenue over the total costs, the productivity for the different factors of production, as the mentioned case of the workforce, will represent the quotient of the sales revenue over

the costs devoted to the specific factor [34]. Of course, the decrease in a factor's productivity will directly affect the global indicator. Productivity is highly regarded by companies, so if a proposed change implies an important increment in the productivity, it will probably be implemented.

The cases for the dismissals of 2 and 3 operators are not as intuitive to evaluate as the previous case. It would be necessary to consider all the factors and, finally, based on the productivity indicators as well as other aspects, a decision could be made.

Despite having considered productivity as a deciding factor, the initial question that was made prior to the evaluation of the reduction of workers in the working shifts was: could the system keep up with the new production pace even though some operators were fired? The answer to this question is that it could, but only for the case of 1 dismissal in the afternoon shift and considering that a drop of not even a 0.5% is not enough to say that "the system does not keep up with the new production pace". The rest of the analysed *Scenarios* would not be able to adapt to this rhythm but, as it has been explained, this does not necessarily imply that the application of the best cases for 2 and 3 dismissals should be directly overruled. Again, productivity will have a big influence in the final decision.

The evaluation of this test scenario number 2 is now finished and, as a final remark, it can be mentioned that an equivalent study could be performed to the model obtained for the test scenario number 1. It will not be done since the evaluations of the test scenarios have focused from the beginning on the real system, but it would be carried out in a similar way to how it has been explained for the real case.

The reason is that it makes no sense to study how a hypothetical situation (the restrictions imposed in test scenario number 2) would affect a hypothetical model (the one considering the new schedule proposed for test scenario number 1). The moment the hypothetical model became the real case, that is, the proposed changes were implemented, it would become sensible to perform the new evaluation (test scenario number 2 to test scenario number 1).

5.3 TEST SCENARIO NUMBER 3

"What if it was considered the replacement of a kneading machine with another one that shortens its processing time?"

The company has been offered the possibility of purchasing a new kneading machine. This new machine would replace one of the two current machines and its most interesting feature is that it reduces the kneading time from the average 15 minutes to an average 10 minutes. Of course, the acquisition of such a device would require a considerable investment. In this case, the price of the machine being considered is \$8,000 USD.

So, it is a matter of analysing if the implementation of the new machine into the current system implies an increase in the total production. If this were the case, then it would be necessary to evaluate if this increase, that would be accompanied by a raise in the sales revenue, is big

enough so that the company is able to recover the initial investment in a relatively short period of time and, right after that, start drawing profit.

The previous is normally evaluated by means of the calculation of the **payback period (PBP)**. This is the time it takes a company to recover an investment. Normally, short payback periods are considered as indicators of worth-making investments. The longer the payback period, the riskier it is to make an investment [35]. It will normally depend on the specific situation of the company analysing the viability on an investment, regarding some aspects like the length of the current working contracts with the suppliers of raw materials or the certainty of the maintenance of the current customers. Then, a company will determine that a payback period is too long if they believe that the risk of a considerable modification in the current state of the company is big. In these cases, the investment would be rejected.

Before computing the payback period it is necessary to obtain the results for the new model, which will be considering the new kneading machine, and find out if the results show that an increase in the production is actually happening. If so, the money that must go inside the formula for the computation of the payback period is the difference of the sales revenues of the new situation and the original one. Given that the sales revenue from the original case is guaranteed, it makes sense to evaluate the increase and not the total sales revenue for the new case in the obtention of the payback period.

$$PBP = \frac{\text{Initial investment (\$)}}{\text{Yearly increase of the revenues (\$/year)}}$$

To apply the modifications for the analysis of this test scenario number 3 it is necessary to modify the parameters of the kneading machine that is wanted to be replaced. In fact, it is just required to change the *Process Time* of the processor modelling the kneading machine. As it was explained previously, in order to take into account the variability of the values for the different times that the machine performs an activity, it was chosen to represent the *Process Time* by means of a statistical distribution, a **normal distribution** with a mean of 900 seconds and a standard deviation of 10 seconds. Keeping the same consideration, the modification will be applied by changing the mean value from 900 to 600 seconds and the standard deviation will be kept at 10 seconds, what will allow for a fluctuation of the times between 9 minutes and a half and 10 minutes and a half.

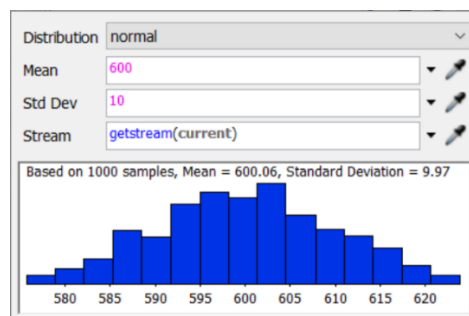


Figure 89

One thing must be made clear, the system is wanted to remain working at a similar pace and with the same situation regarding the raw materials as the one that was obtained as the best

case, or the ideal case, for the model for the real system. That is, the new machine must adapt to the current system and not the other way around.

Going back to the model for the real case, 5125 seconds was defined as the best *Inter-Arrivaltime* for the raw materials. This implies that, daily, the required number of batches of raw material in the warehouse must be of

$$\text{Batches per day available} = \frac{3600 \frac{s}{h} \cdot 24 \frac{h}{day}}{5125 \frac{s}{batch}} = 16.86 \frac{batches}{day} \approx \mathbf{17 batches/day}$$

It is predictable that the system implemented with this testing scenario provides better results for shorter values of the *Inter-Arrivaltime* than the one for the real case. However, in order to fulfil that the machine adapts to the system, the maximum batches per day that can be processed cannot exceed 17, otherwise, it would be necessary to modify the current contract with our suppliers and that option has not been considered.

So, the limit case would correspond to the processing of exactly 17 batches per day. For this, the obtained *Inter-Arrivaltime* equals

$$\text{Inter - Arrivalttime (s/batch)} = \frac{3600 \frac{s}{h} \cdot 24 \frac{h}{day}}{17 \frac{batches}{day}} = 5082.35 s/batch \approx \mathbf{5083 s/batch}$$

So, two *Scenarios* will be analysed by the *Experimenter*: one corresponding to the case expected to be the most interesting and the other for the case that yielded the best results for the model of the real system.

	Variable	Scenario 1	Scenario 2
Variable 1	MODEL:/RawMaterials> variables/interarrivaltime	5083	5125

Figure 90

The results obtained are the following.

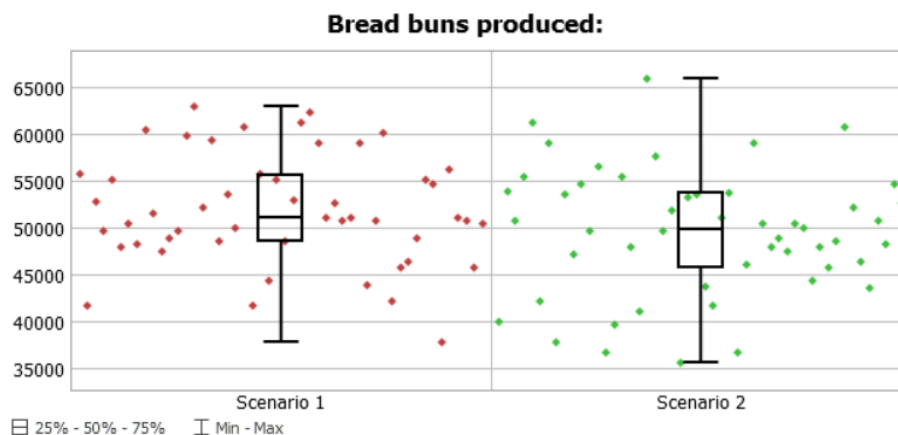


Figure 91

Bread buns produced:						
	Mean (90% Confidence)			Sample Std Dev	Min	Max
Scenario 1	50450	< 51850	< 53249	5876	37740	63000
Scenario 2	47853	< 49468	< 51084	6782	35640	65880

Figure 92

Indeed, the case that was expected to be the most interesting, represented by *Scenario 1*, provides slightly better results than *Scenario 2*. However, it is checked that the application of the new machine to the real system does not provide better results in terms of total production.

Intuition seemed to suggest from the very beginning that this increase was going to happen. A machine that uses shorter times to produce must produce faster and must produce more, or is not it always the case? The total mean production of 3 days obtained in the simulation of the model for the real case was **53650**, whereas for the test scenario number 3 the total mean production barely reaches a value of **51850**. Not only the “expected” raise in production does not happen, but quite the opposite: a decrease bigger than the 3% is occurring.

Of course, it makes no sense to worry about the payback period, since this payback, in the ludicrous case of this investment being made, will never be completed. That is, the initial investment would never be recovered.

Therefore, the running of this test scenario number 3 allows to make clear that an investment of the described characteristics should not be carried out. The system would not benefit at all and some undesired economic losses would be suffered because of the failed investment and the reduction in total production, what would cause a sales revenue drop.

As a conclusion, it can be said that the studied test scenario serves as a great example of how valuable the model for the real system is. It was already mentioned throughout the definition of this model, but for the analysed case it has become a reality: the great deal of relationships that are established within a model make it very difficult to understand or predict its behaviour in the event of certain changes. Probably no one could have predicted that the new system was not going to provide any improvements whatsoever. Probably, lots of people would have thought that this investment seemed reasonable. Yet, this model has been able to test and to show proof about what would be the most sensible decision to be made in this case.

5.4 TEST SCENARIO NUMBER 4

Test scenario number 4: what if it was considered the addition to the system of more machines for the first and second division stages?

The situation is the following: there are up to 3 machines available for each of the processes. Currently, the 2 extra machines are stored, and they are not being used. However, for the test scenario number 4 it is wanted to study if it would provide any benefits to the overall system to incorporate into the process any of the machines.

There are several things that must be taken into account regarding the addition of machines to the current system: firstly, they will require the use of an operator since, as it has already been

explained, these tasks are completed thanks to a combined work of the operator with the machine. This use of operators could compromise the behaviour of the rest of the system and that is why this must be kept in mind. Secondly, operating machinery has a fixed cost by itself. The more machines are put to work, the bigger the cost. So, it might happen that the sales revenue increases with the introduction of new machines, but if this increase is exceeded by the raise in the machinery cost then it is a change for the worse.

The tool that will be used to perform this analysis will be the optimiser *OptQuest* that, as it has been mentioned earlier, is a very powerful optimisation package that is integrated with *FlexSim*. It could have been used earlier, but the evaluations have been carried out using other equally valid alternatives. However, for the specific scenario being tested now it will be shown why it seems to be the most reasonable idea to choose *OptQuest* as the solving tool.

In the latest versions of *FlexSim* (the one being used for this project is *FlexSim 2020 Update 2*) the optimiser tool has started to be included as an additional part of the *Experimenter* rather than being an individual tool by itself. This measure has helped to show how important the combination of simulation and optimisation is. They are no longer understood as two independent concepts, but two ideas that must work together for the overall benefit of the system being studied.

Therefore, it is accessed through the *Experimenter*, and the 3 tabs that will be used for the definition of the parameters strictly related with the optimisation are the following.

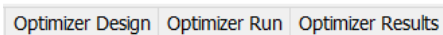


Figure 93

The variables that will be considered by the *Optimizer* must be introduced the same way as they were introduced for the running of *Scenarios* by the *Experimenter*. Simulation (represented by the *Experimenter*) and optimisation (represented by the *Optimizer*) are now integrated, and that is why a single definition of the variables is enough for the completion of the two different analysis.

Also, the defined *Performance Measures* will be available for the evaluation of either the simulation or the optimisation.

So, it is time to determine which are precisely those variables and *Performance Measures* of interest for the definition of the optimisation process, what will be performed inside the *Optimizer Design*. It is important to remember that the test scenarios are being built by means of the application of several changes to the original model, so it might occur that some of the variables and *Performance Measures* are already defined.

This is the case for the variable that refers to the *Inter-Arrivaltime*, that will still be considered for the optimisation, and the *Performance Measure* which is defining the total output of the system (*Bread buns produced:*), whose value will definitely be very relevant for the analysis.

Two new variables must be defined so that the possibility of working with 1 to 3 machines in each of the mentioned stages is studied. To do so, similarly to how it was done in the test

scenario number 2, *Groups* will be created. *Group1* will include the separator in charge of the first division, which is *FirstDiv*. The same will happen for *Group2*, that will include the separator in charge of the second division, or *SecDiv*. The variable option *Nr Objects in Group* must be selected for each of the cases and then, all the elements are ready for the definition of the *Optimizer Design*.

	Variable
Interarrivaltime	MODEL:/RawMaterials>variables/interarrivaltime
FirstDiv	Nr Objects in Group1
SecDiv	Nr Objects in Group2

Figure 94

This step allows the determination of the following: firstly, in *Variables*, there must be specified a *Type* for the already defined variables, which will be *Integer* for all the variables, and it must be stated the *Lower bound* and *Upper bound* for each of them. This way, it is possible to limit the options for the optimiser to randomly select points within these intervals in order to figure out the best possible solution. The *Interarrivaltime* will be considered from 5083 to 5100 seconds. Keeping the same reasoning regarding the availability of raw materials as it was explained in test scenario number 3, the limit case is being studied and, despite 5125 seconds being the value that yields the best result for the real case, this *Upper bound* has been reduced to 5100 in order to obtain a considerable decrease in terms of computational cost and, as it was mentioned previously in this paragraph, to help the optimiser by providing a shorter list of possible points to choose for their evaluation. Regarding *FirstDiv* and *SecDiv*, the intervals will go from 1 to 3.

Variables					
	Type	Lower Bound	Upper Bound	Step	Group
Interarrivaltime	Integer	5083	5100	N/A	N/A
FirstDiv	Integer	1	3	N/A	N/A
SecDiv	Integer	1	3	N/A	N/A

Figure 95

Secondly, *Constraints* can be added to the optimisation conditions. For example, if it had been stated that a maximum total 4 machines can be employed between both stages, some expression like the following should have been written: “[*FirstDiv*] + [*SecDiv*] <= 4”.

Finally, in *Objectives*, the *Objective Function* must be defined. There might be more than one, and it represents the expression which is wanted to be maximised or minimised. That is, all the effects that are wanted to be taken into account must be appropriately introduced in the *Objective Function* so that the final results provided by the optimisation succeed in answering the question that this function is posing.

For this case, it has been decided that the *Objective Function* will be the total income, considering the sales revenue minus the working cost of the machines. The following values have been considered: the sale price of each of the bread buns is **\$0.1**, and the daily working cost of the machines for each of the stages is **\$3**, for the *FirstDiv*, and **\$7**, for the *SecDiv*. This is how it should be introduced.

Objectives			
		Function	Direction
	Objective 1	[Bread buns produced:]/10 - 9*[FirstDiv] - 21*[SecDiv]	Maximize

Figure 96

Since the computations are being performed for the total production in 3 days, the working costs of the machines have been multiplied by 3.

Once the *Optimizer Design* is finished, it is time for the definition of the conditions for the *Optimizer Run*. The optimisation will run for the 3 days that have been analysed in all the previous cases. A maximum value of 100 solutions has been set, with a *Wall Time* of 0.00. This refers to the real time that the optimiser spends finding solutions. By establishing this value at 0.00, the optimisation will not stop after a certain amount of time but after the established *Max Solutions* are reached. Also, several replications will be run for each of the solutions, in this case, 20. This way, the already known variability of the system will be taken into account.

Optimize

End Time: 5:00:00 | 09/10/2020

Run Time: 259200.00 | Seconds

Warmup: 0.00 | Seconds

Wall Time (seconds): 0.00

Max Solutions: 100.00

Show advanced options

Run multiple replications per solution

Manual stop only

Figure 97

By clicking *Optimize*, the optimisation begins. These are the results that are obtained after the optimisation process has been completed.

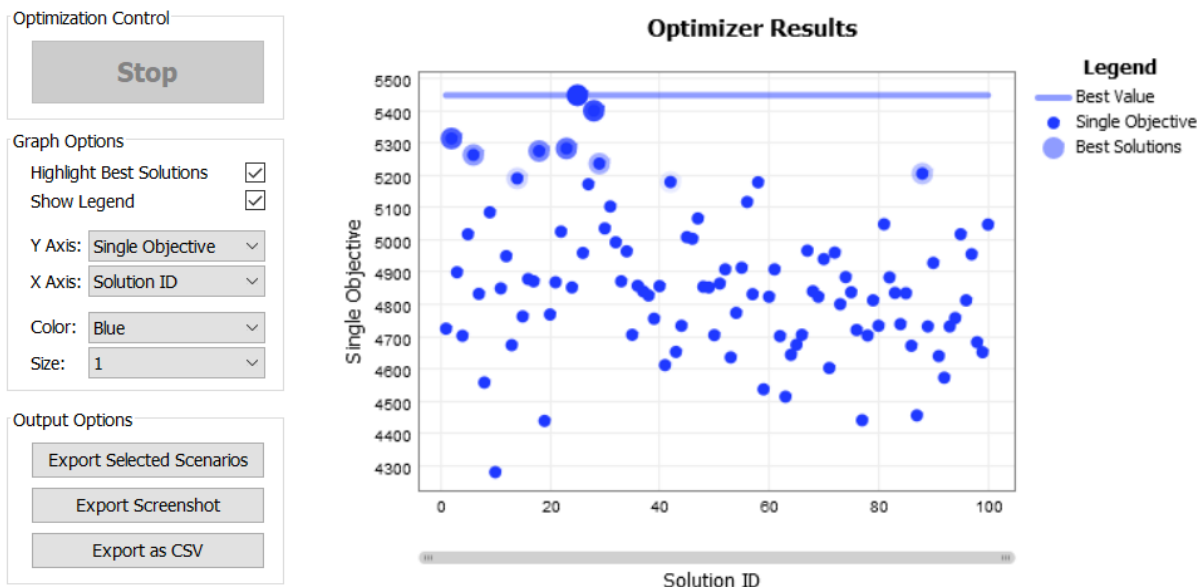


Figure 98

By looking at the *Graph Options* it can be easily deduced what results are being represented. In this case, the *Single Objective*, which corresponds to the only *Objective Function* that has been defined, is being graphed versus the *Solution ID*, a value that is numbering the solutions from 1 to the total 100 that have been obtained. The best solution is indicated by the *Best Value* line

and corresponds to a value for the *Single Objective* of **5448** and the *Solution ID* is **25**. That is, solution number 25 suggests that the maximum possible income, considering as such the defined *Objective Function*, is \$5448.

By selecting different *Graph Options* it can be seen what values for the variables are the ones that have provided the best solution in terms of income. These are the results for each of the three variables.

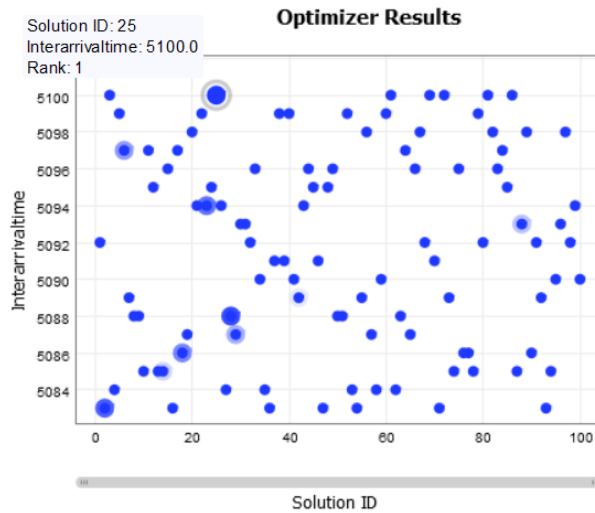


Figure 99

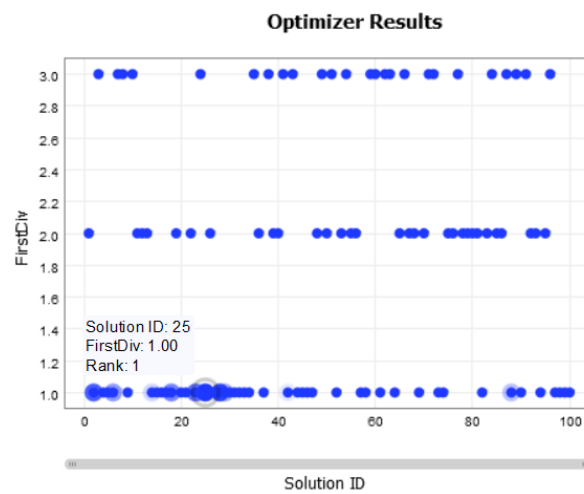


Figure 100

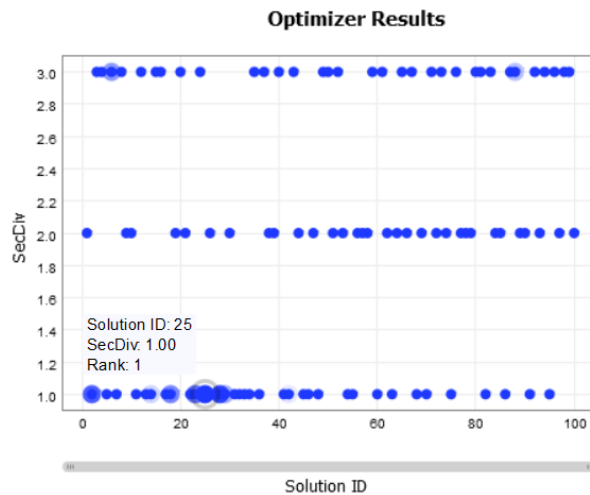


Figure 101

The best solution is obtained for an *Interarrivaltime* of 5100 seconds, and for 1 machine in each of the division stages. The number of produced buns for this solution must have been equal to 54780 since if the cost of the machines, which has been \$30 because there is only 1 machine per stage, is added to the total income, the sales revenue can be obtained. A sales revenue of $\$(5448 + 30) = \5478 implies that $5478/0.1$ bread buns must have been sold, which is equal to **54780** indeed. It can be checked, by selecting the appropriate *Graph Options*, that this value is correct.

In fact, the best solution does not only provide the best income amongst the 100 solutions, but it also provides the best result regarding bread production (54780). The option of adding more machines to the stages was already assumed to be more expensive, in terms of machine cost, than the existing system. However, it was hoped that the implementation of machinery would incur in a big enough increase in terms of bread production, so that the raise in the machine cost could be compensated. This increase does not occur. Probably, the system, as it is configured currently, does not allow the addition of extra machinery. The existing balance within the process seems to be altered by these changes and, as a result, the production is not enhanced but worsen.

As an example of how this optimisation is ranking its solutions based on the *Objective Function* it is interesting to observe the best solutions for the cases that are not considering 1 machine per stage. They are remarkable the cases of *Solution ID: 6*, ranked 6th; and *Solution ID: 88*, ranked 8th. Both solutions are considering the implementation of 1 machine for the first division stage and 3 machines for the second division stage. So, they will definitely destine more money to the machinery than the cases with 1 machine per stage, what could be a pitfall regarding income. As a matter of fact, these two solutions provide better results in terms of bread production than some of the solutions ranked above them, but the machine cost is determining that they are less interesting considering income.

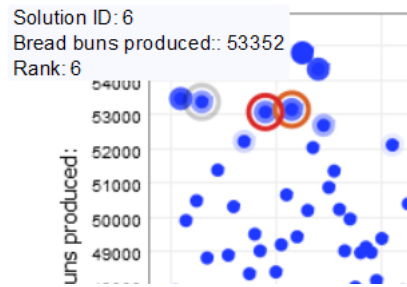


Figure 102

For example, the selected solution is obtaining a greater value for its production (53352) than the other two values, circled in orange and red. They are ranked 4th and 5th, respectively, and their values for the *Bread buns produced*: are 53130 and 53057. However, these two solutions are using 1 machine per stage, what represents a big enough difference in terms of machine cost to define them as better solutions than *Solution ID: 6*.

To finish, it must be highlighted the great importance of having used the *OptQuest* option for the obtention of the results for the test scenario number 4. It is been made clear how powerful this tool is, and it is not surprising, therefore, that this tool is so widely employed in the industrial world.

Also, and similarly to how it happened in the test scenario number 3, the solution is recommending that the proposed changes are not applied to the system. Again, it could have been intuitive to believe that the addition of machinery was a guarantee of success. Yet, the resolution of this test scenario has provided evidence to deny that.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

- 1) Through the explanation of the different parts and stages of the bread production process under study, it has been achieved a quite detailed description of the overall production system. This allows the reader a better understanding of the different steps of the complete process and how these parts interact with each other.
- 2) By means of the *FlexSim* software, it has been possible to deal with the data of a real company, such as the one studied in [25], and to do that in a smart way since *FlexSim* has displayed its ability to “translate” this “real world’s” information into “virtual world’s” intelligence, whose handling is straightforward for the software.
- 3) An introductory theoretical model has been built and has succeeded in providing equivalent results to the ones obtained with the resolution of the proposed Assembly Line Balancing Problem (ALBP). Moreover, this model has set the starting point for the building of the rest of the models in this project.
- 4) From the previous model, two more models have been built: the first, a deterministic model, has implemented certain changes to the theoretical model, in order to represent the real system being modelled more accurately. The results provided by this model have turned out to be really good. However, the fact that this model only uses deterministic data implies that it is not fully realistic and, therefore, its results are not truly relevant.
- 5) Trying to improve the previous model, the deterministic model has been made stochastic: this second one has taken into account the time variability of the processes that compose the full system, what has helped to achieve a more realistic model. At this point, the process of further developing the original model has stopped, and this last model has been considered as the final one. The results provided by its simulation suggest that, since the difference between them and the expected ones is reasonably small, the model can be considered as valid. Therefore, it can be said that the building of the simulation model has been successful.
- 6) Once the final model has been obtained, it has been used for the evaluation of several test scenarios that have been proposed: *Test Scenario number 1* proposed a change regarding the working schedule and its evaluation has confirmed that it would be beneficial if this change were implemented. *Test Scenario number 2*, on the other hand,

presented a hypothetical situation where the event of some trouble concerning the supply of raw materials was evaluated. In this case, the option of making operators redundant was analysed and the possibility of considering 1 dismissal has seemed to be reasonable, according to the results. *Test Scenarios number 3 and 4* both proposed changes regarding the employed machinery in different stages of the production system. In both cases, the evaluation of the scenarios has suggested that the changes are not a good idea, since they do not represent a clear improvement to the current system. Also in both cases, the simulation has been able to detect that the proposed changes were not desirable, despite the fact that intuition seemed to suggest the opposite. Actually, these proposed 4 scenarios have just served as an example of the importance of the obtained tool. The main value of the model analysing these scenarios is that it could be asked to analyse any possible scenario that might be thought of, and, since it has proven to be a valid model, the results provided by the simulations would be equally valid and of great interest and value for the analyst performing the study.

- 7) Ultimately, this project has succeeded in attaining its principal goal, since a fully capable, valid, and extremely useful tool has been developed, for evaluating how a real company performs currently and for being able to predict the modifications of its behaviour as a consequence of any changes occurring. Therefore, as it has been shown with the example of the 4 test scenarios, any company working with some tool like the one that has been created in this project will be able to make sensible and reasoned decisions regarding the implementation of changes to the system to optimise its overall performance.

6.2 RECOMMENDATIONS

One of the main features of this project is that it could be further developed endlessly. That is, the model that has been chosen as the final one could be improved by carrying out a deeper analysis of the real production system and by implementing as many changes as considered in the virtual model. The more thorough the real system's data collection and analysis, the more exact the updated model and, therefore, the more accurate the results.

So, as the principal recommendation for a further study taking this project, or a similar one, as the starting point it could be highlighted the necessity of a continued supervision, research, and modification of the features of the virtual model so that it follows the exact same conditions and alterations occurring to the real system at all times. Of course, this is the ideal case, and it is very difficult to reach but, for the team in charge of this task in a company, it must be its primary goal.

For example, in this particular case, the obtention of real data regarding the production system's current situation regarding **suppliers** and **customers**, which has not been available for the completion of this work, would directly imply an enormous benefit to the truthfulness of the model. This would add to the process several additional steps that should be modelled and that would probably have an influence on the behaviour of other parts within the system.

Also, technology should be used in a smart way to try to monitor and standardise as many variables and parameters as possible in the different stages. In today's industry, as it was mentioned in the introduction of the concept of Industry 4.0, communication between machines on their own is possible, through IoT (Internet of Things). So, it would be desirable to use sensors and intelligent devices to establish a flow of information from the machines and their "conversations" toward the simulation software and the other way around. This way, it becomes possible to keep track of all the changes occurring in the machines and, also, to directly implement some modifications in the system that have been previously checked with the simulation software. A fully automated and smart system will have the best possible virtual model and, therefore, the tool will be the most powerful it can be.

7. BIBLIOGRAPHY

- [1] J. C. Capel, *El pan*. Madrid: Montserrat Matéu, 1991.
- [2] I. Tejera, *El libro del pan*. Madrid: Alianza editorial, 1993.
- [3] G. Varela, *El pan en la alimentación de los españoles*. Madrid: Eudema, 1991.
- [4] C. Fernández, "Historia del Pan," *CEOPAN* [Online]. Available: <http://www.ceopan.es/index.php?type=public&zone=smartportalcategorias&action=view&categoryID=295&codeID=295>. [Accessed: Nov 27, 2020].
- [5] Comunicación, "Historia de la Evolución del Pan hasta hoy," *MUNDOPAN*, 2016 [Online]. Available: <https://mundopan.es/historia-de-la-evolucion-del-pan-hasta-hoy/#:~:text=Las%20primeras%20evidencias%20arqueol%C3%B3gicas%20del,molidos%20a%20mano%20con%20piedras>. [Accessed: Dec 28, 2020].
- [6] J. Calaveras, *Tratado de Panificación y Bollería*. Madrid: AMV Ediciones, 1996.
- [7] Dulcypas, "Métodos de amasado," *pasteleria.com DPAS*, 2000 [Online]. Available: <https://www.pasteleria.com/articulo/200002/1552-metodos-de-amasado>. [Accessed: Dec 28, 2020].
- [8] A. Maria and W. Whitt, "Introduction to Modeling and Simulation," in *Winter Simulation Conference*, 1997, vol. 1, pp. 7-13.
- [9] T. Altiok and B. Melamed, *Simulation Modeling and Analysis with ARENA*. USA: Academic Press, 2007.
- [10] R. Cao, *Introducción a la Simulación y a la Teoría de Colas*. A Coruña: NETBIBLO, 2002.
- [11] E. García, H. García and L. E. Cárdenas, *Simulación y análisis de sistemas con ProModel*. México: Pearson Educación, 2006.
- [12] O. J. Herrera and L. A. Becerra, "Diseño General de las Etapas de Simulación de Procesos con Énfasis en el Análisis de Entrada," in *Twelfth LACCEI*, Guayaquil, Ecuador, July 22-24, 2014.
- [13] T. K. Abdel-Hamid, "THE DYNAMICS OF SOFTWARE DEVELOPEMENT PROJECT MANAGEMENT: AN INTEGRATIVE SYSTEM DYNAMICS PERSPECTIVE," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1984.
- [14] A. A. Tako and S. Robinson, "The application of discrete event simulation and system dynamics in the logistics and supply chain context," *Decis. Support Syst.*, vol. 52, no. 4, pp. 802-815, 2012 [Online]. Available: <http://dx.doi.org/10.1016/j.dss.2011.11.015>. [Accessed: Dec 29, 2020].

- [15] A. A. Tako and S. Robinson, "Model building in System Dynamics and Discrete-event Simulation: a quantitative comparison," 2008 [Online]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.408.5655&rep=rep1&type=pdf>. [Accessed: Dec 29, 2020].
- [16] J. B. Jun, S. H. Jacobson and J. R. Swisher, "Application of Discrete-Event Simulation in Health Care Clinics: A Survey," *J. Oper. Res. Soc.*, vol. 50, no. 2, pp. 109-123, Feb 1999 [Online]. Available: <https://www.jstor.org/stable/3010560>. [Accessed: Dec 29, 2020].
- [17] "FlexSim," [Online]. Available: <https://www.flexsim.com/es/flexsim/>. [Accessed: Dec 29, 2020].
- [18] "Simulación de un proceso industrial mediante el software FlexSim," class notes for 37810, Master's Degree in Automatics and Robotics, Univ. Alicante, Feb 6, 2012 [Online]. Available: https://rua.ua.es/dspace/bitstream/10045/20587/1/Simulacion_de_un_proceso_industrial_mediante_FlexSim.pdf. [Accessed: Dec 29, 2020].
- [19] R. Casadiego, *Guía de Usuario para el modelamiento y análisis con el software Flexsim*. Cúcuta, Colombia: Group of Investigation in Productivity and Competitiveness, Univ. Francisco de Paula Santander.
- [20] Descreye Solutions, "Best Discrete Event Simulation Software - 2019," OPS, Sep 9, 2019 [Online]. Available: <https://www.descreye.com/blog/a-comparison-of-discrete-event-simulation-software-2019/>. [Accessed: Dec 29, 2020].
- [21] SAP Insights, "What Is Industry 4.0?," SAP [Online]. Available: <https://insights.sap.com/what-is-industry-4-0/>. [Accessed: Dec 29, 2020].
- [22] E. Díaz, "Industria Conectada 4.0: el motor del cambio económico," *Blogthinkbig.com*, Sep 27, 2017 [Online]. Available: <https://blogthinkbig.com/industria-conectada>. [Accessed: Dec 29, 2020].
- [23] "Meet the Three Industrial Revolutions," *Trailhead* [Online]. Available: <https://trailhead.salesforce.com/es-MX/content/learn/modules/learn-about-the-fourth-industrial-revolution/meet-the-three-industrial-revolutions>. [Accessed: Dec 29, 2020].
- [24] W. Bolton, *Programmable Logic Controllers (6th Ed.)*. USA: Newnes, 2015.
- [25] L. C. Vásquez, "PROPUESTA DE MEJORAMIENTO DE PROCESOS EN EL ÁREA DE PRODUCCIÓN DE LA EMPRESA PANIFICADORA PANARTE A TRAVÉS DEL ESTUDIO DE TIEMPOS Y MOVIMIENTOS," M.S. thesis, EPN, Quito, Ecuador, Apr 2017.
- [26] "Bread Making Process - A Guide for the bread maker," *Food Crumbles*, Apr 18, 2018 [Online]. Available: <https://foodcrumbles.com/bread-making-process-a-concise->

- overview/#~:text=%20Bread%20Making%20Process%20%E2%80%93%20A%20Guide%20for%20Step%209%3A%20Cooling%20%26%20Eating%20More%20. [Accessed: Dec 29, 2020].
- [27] D. L. Nelson and M. M. Cox, *Lehninger Principles of Biochemistry (4th Ed.)*. New York: W. H. Freeman and Company, 2005.
- [28] The Editors of Encyclopaedia Britannica, "Fermentation," *Encyclopaedia Britannica*, May 6, 2020 [Online]. Available: <https://www.britannica.com/science/fermentation>. [Accessed: Dec 30, 2020].
- [29] L. Alba-Lois and C. Segal-Kischinevzky, "Yeast fermentation and the making of beer and wine," *Nature Education*, vol. 3, no. 9, pp. 17, 2010 [Online]. Available: <https://www.nature.com/scitable/topicpage/yeast-fermentation-and-the-making-of-beer-14372813/>. [Accessed: Dec 30, 2020].
- [30] "How the Fermentation Process Works - Yeast & Its Byproducts," Food Crumbles, Jan 3, 2016 [Online]. Available: <https://foodcrumbles.com/fermentation-process-beer-bread-yeast/>. [Accessed: Dec 29, 2020].
- [31] I. Pascual, "Metodologías de resolución para el problema simple de equilibrado de líneas de montaje," undergraduate thesis, Univ. Seville, Seville, Spain, 2015 [Online]. Available: <https://idus.us.es/bitstream/handle/11441/38380/Pascual%20Garc%C3%ADa,%20Ignacio.pdf?sequence=1>. [Accessed: Dec 30, 2020].
- [32] K. B. Zandin, *Maynard's Industrial Engineering Handbook (5th Ed.)*. New York, NY, USA: McGraw-Hill, 2001.
- [33] M. D. Ugarte, A. F. Militino and A. T. Arnholt, *Probability and Statistics with R (2nd Ed.)*. CRC Press, 2015.
- [34] J. Martínez, "5. PRODUCTIVIDAD DE UN FACTOR Y PRODUCTIVIDAD GLOBAL," *ECONOSUBLIME*, Nov 4, 2020 [Online]. Available: <http://www.econosublime.com/2019/04/medicion-productividad.html>. [Accessed: Dec 30, 2020].
- [35] G. Reniers, L. Talarico and N. Paltrinieri, "Chapter 16 - Cost-Benefit Analysis of Safety Measures," in *Dynamic Risk Analysis in the Chemical and Petroleum Industry*, Butterworth-Heinemann, 2016, pp. 195-205 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780128037652000160>. [Accessed: Dec 30, 2020].