Facultad de Ingeniería
INGENIERÍA INDUSTRIAL
Trabajo de Grado – Primer Semestre 2019

Capstone Final Project

[183024] Simulation-Optimization model for a Hybrid Flow shop. Case Study, Chemical Industry.

Maria José Orozco Quiroz[a,c] , María Alejandra Peréz Uribe[a,c] ,
Juan Fernando Díaz Guerrero[a,c], Nicolás Duque Minaya[a,c]
Jose Fernando Jiménez[b,c]

[a]*Undergraduate students, Industrial Engineering*
[b]*Associate Professor, Project Director*
[c]*Pontificia Universidad Javeriana, Bogotá, Colombia*

Abstract

Currently, operations research techniques have proved to make a significant impact in modern manufacturing systems as it provides an enhanced productivity and performance on highly competitive markets. Hybrid Flow Shop Problem, also known as Flexible Flow Shop problem, is a scheduling problem related to a group of parallel machines per stage, frequently associated with time minimization in a manufacturing environment. This problem is considered a NP-hard problem due to the combinatorial decisions and the demanding computing resources and execution time in its resolution. This research is focused on a case study of Fuller Pinto which is an international Colombian-based company from the chemical industry that presents a Hybrid Flow Shop environment on its shop-floor. Even today, the company still have issues associated to late deliver of products due to poor scheduling, causing the planned production no to be completed. Moreover, these unfinished products become backorders with higher importance that must be supplied mandatory. For this reason, this research proposes a decision support system based on a simulation-optimization model for Fuller Pinto scheduling, aiming the minimization of the *total weighted tardiness*. The proposed model features enhanced capabilities simulating the behavior of the complex manufacturing environment and hybridize iteratively an optimization algorithm to support the Fuller Pinto decisions. To validate this model, different scenarios or instances related to the production behavior of chemical products have been tested. Comparisons among the proposed simulation-optimization model, the SPT dispatching rule and historical data of the company are presented. As results, the model provides an optimal schedule of jobs for each campaign of Fuller Pinto.

Keywords: Hybrid Flow Shop, Flexible Flow Shop, optimization, scheduling, simulation-optimization, dispatching rule.

1. Justification and problem statement

Fuller Pinto is a chemical company dedicated to the production of products for personal care, home cleaning agents and industrial cleaning agents, such as disinfectant cleaners, floor wax, liquid soap, antibacterial gel, bleaches, detergents, degreasing, brooms, brushes and mops, among others. The manufacturing shop-floor of Fuller Pinto have the following production machinery and workstations: injection area, insertion area, microfiber area, screen printing area, turning and mechanics area, wood processing area and chemical area.

The current performance of the company has allowed to maintain his presence in the national and international market, supplying to countries like Peru. However, Fuller Pinto has certain problems that threatened the company productivity and efficiency. Those problems involve manpower issues, material waste, and tardiness associated to the late delivery of products. What is more, in the supply and production department, voluntary retreats frequently associated with better job offers, disagreement with superiors and complains because of the salary and working conditions are constantly presented. In areas such as injection and inserted, material waste is presented due to defective product reprocessing. In addition, one of the major problems of the company is associated with the production decision making in chemical area, since it is still done in an intuitive way and constantly learning from experience. The current tardiness problem occurs when the completion time of the products in chemical area goes beyond the due date stablished. Therefore, a lot of products become delayed because of the lack of expertise and not well-suited scheduling methods. The previous statement contributes significantly to the decision-making problem in chemical area. As a result, the decision-making problem leads to bigger issues such as a supply troubles and sales problems.

This dissertation focuses in the scheduling method performed in chemical area. This important area contributes around 80% of Fuller Pinto's total sales. Consequently, this area has strict production and supply policies that must meet the company goals. Based on the above, it is expected that an intervention in the scheduling system and it decision making method may improve the productivity and efficiency expected in the chemical area.

The chemical area of Fuller pinto has the following production policies:

- The commercialization of the products of the chemical area is carried out through the traditional channel, named *Catalogue sale.*
- The catalogues, designed and developed by the management area, are distributed monthly to the chemical area manager and each catalogue contains two campaigns for sales and production.
- **At Fuller Pinto,** a **campaign** refers to the release of products in the catalogue that are available for sales during a period of **20 days**. Manufacture and chemical area must prepare and produce the product demand requested for these campaigns.
- Fuller Pinto has a productivity indicator to measure the chemical area production performance as a percentage. This performance measure is known as **supply percentage** and refers to the amount of finished product at the beginning of the 20 days of the campaign that will be offered to the public in order to supply the requested orders. The chemical area policy is to ensure at least a 98% of the supply percentage. Then, the fulfillment of this percentage must be guaranteed just before the start of the campaign.
- Checking the supply percentage: It consists on the verification of the number of finished products at the end of a campaign, respect to those requested in the catalogue for that campaign.

The manufacturing shop-floor for chemical area consist of two stages: the mixing stage and the packaging stage. To carry out the production there is a set of machines and/or tools: 23 mixing tanks/storage located in the first stage, which can be used according to the product properties (viscous or liquid). In addition, the shop-floor has five machines for packaging located on the second stage. The processes in this area is as follows. First, the products ordered on each campaign are poured, mixed and temporarily stored in tanks of the mixing stage. Depending on the product and the quantity, this process can last between two and eight hours in production. Once the product is ready for packaging, these are transferred through hoses and pipes to the packaging machines on second stage. In addition, the packaging processing time depends on the product type, its volume and the machine that will package it. Finally, after the packaging, the product is sealed, labeled and codified for transportation to the finished products storage area. Figure 1 illustrates the shop-floor for the chemical area.

It is important to note that the use of the tanks is restricted to the properties of the product (viscous or liquid) and, for the normal flow of product, the human operator must decide and prepare the respective machine that will receive and package the product. In addition, the packaging machines are subject to the types of raw material that can be processed and to the volume of the container where the item is packaged. For instance, while the PQEV2PIS machine and the PQEVM201 only pack viscous products, the PQELDIMA and PQEMSEMI package strictly liquid products. In some cases, manual work is required with liquid products. For this reason, the flow of each product through the shop will depend on the characteristics it has to store, mix and package.

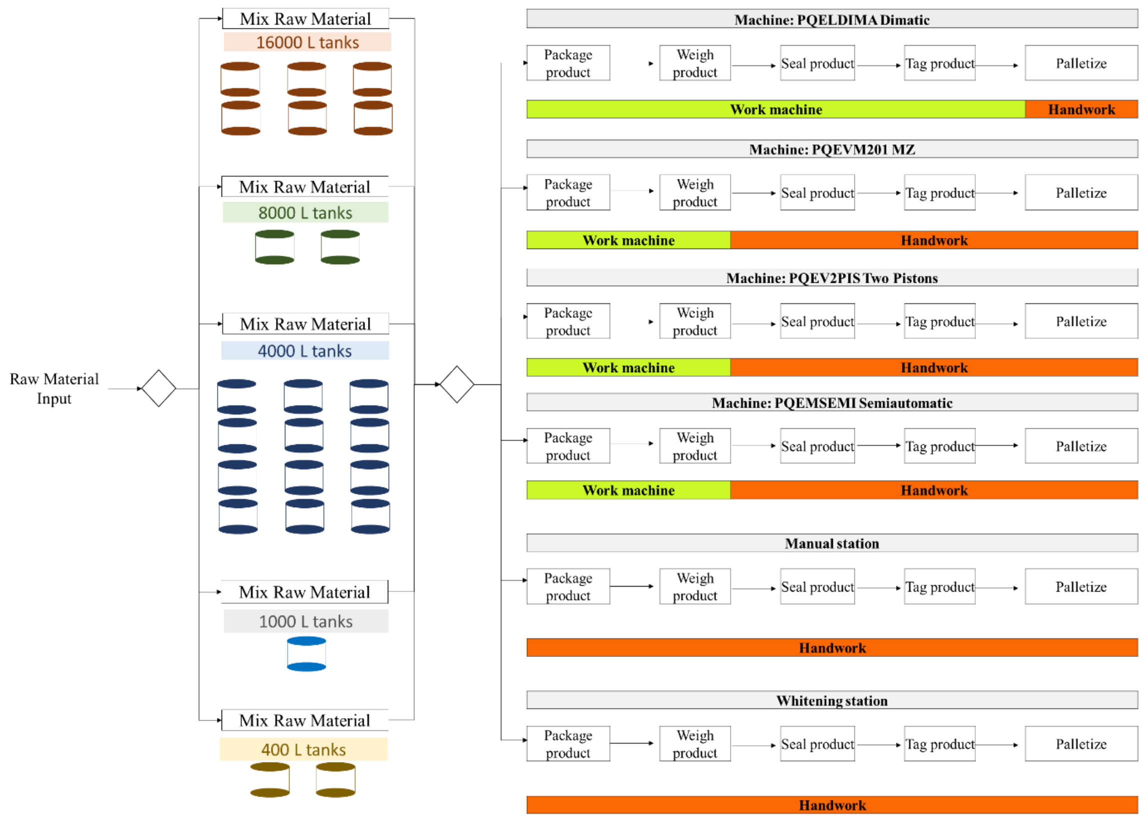


Figure 1. Block Diagram - Chemical Area

Given the above information, it can be described the decision-making process of the company. At the beginning of a cycle, the production manager receives the catalogue and verifies the finished product inventory to supply the distribution centers, in other words, checks the supply percentage. Once the supply percentage is checked, jobs of the previous campaign with a lower percentage are given higher priority, as they are the ones that must be produced in greater quantity. The goal of the chemical area is to ensure that at least the supply percentage at the start of a campaign is equal to 98%. To achieve this, the production manager identifies two different moments in the current campaign, the Sales Cycle and the Production run shown at the bottom of Figure 2. During a period of 20 days there is a campaign that goes on sale and it is represented by Sales Cycle top arrow. It means that the products that belongs to that campaign are now on the marketplace. On the other hand, there are other products that are going to be produced during the 20 days and are represented by Production run arrow. The two cycles mentioned above happen at the same time, and once the 20 days are run out the campaign products that have being produced are now on the Sales Cycle because the company needs to supply their different shops. Ideally, the campaign production orders to be sold on t needs to be ready at least 98% at the end of the t - 1 campaign cycle. Thus, the catalogue order of the $t + 1$ campaign must be covered by at least 98% in the cycle associated with the t campaign.

In order to demonstrate the situation described above, Figure 2 shows the production policy based on campaigns in the chemical area. In this figure, it is observed that if the minimum condition of the supply percentage is not fulfilled, this missing percentage is considered as backorders that accumulate in the following campaign.

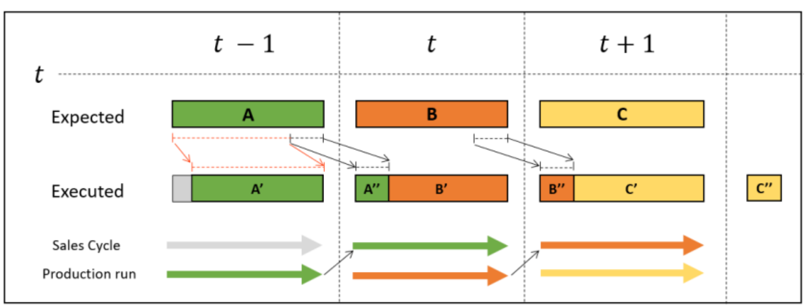


Figure 2. Behavior over time of campaign-based production policy

In addition, Figure 3 presents the supply percentage that the last seven campaigns had, from the month of June to October of the year 2018.

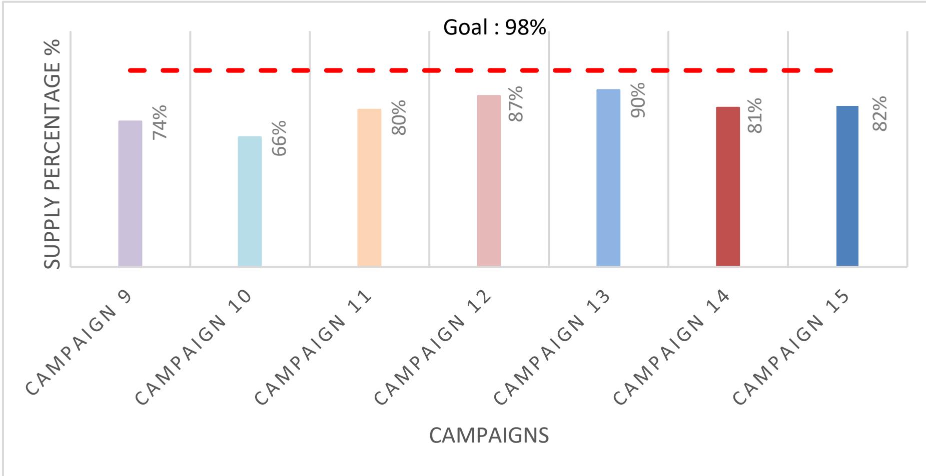


Figure 3. Campaigns Supply Percentage

Figure 3 illustrates the non-fulfillment of the supply goal at the beginning of each campaign is visible, since none exceeds or equals 98%. This percentage difference of the target in a few months increases the production time needed to meet the demand associated with campaigns. This also generates overtime work, and additional costs of wages. Furthermore, this non-fulfillment of the supply, results in lost sales due to unavailability of finished product, as well as losses due to penalties and breaches of contracts.

In conclusion, this research is focused on the scheduling of the production tasks of Fuller Pinto's chemical area, aiming to increase the supply percentages of campaigns. In this sense, the planning and scheduling of the work requested in each campaign, would allow the chemical area to adequately organize its processes with the aim of supplying them before the end of a campaign. For this reason, the minimization of the tardiness in the work ordered in the catalogues in the cycle of each campaign is the goal of the development of this project.

2. Literature Review

Currently, the chemistry area of the Company Fuller Pinto requires decision-making through all the shop-floor. First, the distribution of raw material among the 23 mixing tanks located in parallel must be done. These clustered tanks in the mixing area belongs to the first stage in the shop. The second choice is related to the packaging machines selected to be used in the second stage of the shop. Furthermore, the transportation time between the first and second stage is negligible because of the powerful pumping of fluid throughout the pipes. The setup time is a considerable restriction owe to the cleaning tasks that must be done when the fluid changes from liquid to viscous. It is no possible for a job to skip a stage; they must go through both stages. In addition, there is no precedence constraint considered in the production area. Finally, each of these machines can process a work at a time. Therefore, the production environment of Fuller Pinto is considered flow shop hybrid or hybrid flow shop (HFS).

2.1. Hybrid Flow Shop Problem (HFSP)

The problem of flow shop hybrid or hybrid flow shop problem (HFSP), describes a manufacturing environment, including a number of stations that have a set of M machines (Pinedo 2012). Machines, processes N products and are located in parallel at each station. In addition, the buffers existing between the production of successive processes are unlimited usually when small products are produced as printed circuits (Aurich, Nahhas, Reggelin, & Tolujew, 2016). They are considered NP-Hard and found in real life in textiles, paper and electronics Industries (Ruiz & Vásquez,2010).

The objective function of a HFSP is often associated with time minimization. Specifically, the function to minimize is associated with end-up times of the last operation or Makespan (Aurich et al. 2016), maximum tardiness or maximum lateness (Fakhrzad & Heydari 2008) and Flow Time (Lin & Chen 2015) for the cases studied. However, the functions can also be associated with the minimization of average or total weighted completion time and average maximum tardiness or total weighted tardiness (Pinedo, 2012). The mathematical programming model developed by (Ruiz et al. 2008) was considered to model the shop-floor environment of chemical area in Fuller Pinto. The study made by (Ruiz et al. 2008) is very useful to this research because of the complexity of the production environment he investigated. Some constraints considered by (Ruiz et al. 2008) include the sequence depend setup times and machine eligibility, that either apply to chemical area shop-floor environment.

A literature review was conducted on Hybrid Flow Shop Problem (HFSP). The databases consulted were: IEE Explore, ProQuest, ScienceDirect, Emerald Insights, Taylor & Francis Group and Academic Google. The keywords for this review were: hybrid flow shop, flexible flow shop, scheduling, unrelated parallel machines, sequence-dependent setup times, Branch & bound HFSP, heuristics HFSP, Metaheuristics HFSP, Simulation-optimization). The inclusion criteria were the year of publication (from the year 2000 onwards), HFSP or flexible flow shop applied to manufacturing industries and the method developed in the research. The methods admitted were exact methods algorithms, Heuristics, Metaheuristics, Simulation-Optimization, and Mixed Integer Programming with objective functions related to minimizing the Makespan, the Tardiness or the Flow Time. The exclusion criteria were Flow Shop problem (FSP), Job Shop problem (JSP) and any environment other than HFS problems were removed.

	a				b										c				d	e
a General solution method **b. Alternative solutions** **c. Objective funciton** **d. Additional Considerations** **e. Result**	Exact method algorithm	Heuristic	Metaheuristic	Simulation-optimization	Branch&Bound	Dispatching rules	Tabú Search(TS)	Immume Algorithm (IA)	Neural Networks (NN)	Simulated Annealing (SA)	Genetic Algorithm (GA)	Simulation-optimization	Optimal computing Budget allocation (OCBA)	Mixed integer programming (MIP)	Minimize end time of last task or makespan	Minimize total tardiness	Minimizeflow time	Others	**Additional Considerations**	**Result**
(Carlier & Néron, 2000)	•				•										•				Branching proposal	Minimize makespan
(Moursli & Pochet, 2000)			•								•				•				Proposal for a new decoding	Minimize later time
(Zandieh, Fatemi, & Moattar, 2006)			•	•				•			•				•				setup times	Compare effectiveness between algorithms
(Low, Hsu, & Su, 2008)		•				•									•				Use of two stations	Compare combinatorias of sequencing rules and dispatch rules
(Fakhrzad & Heydari, 2008)		•					•			•						•			Heuristic proposal	Compare results Taboo search and simulated annealing
(Chaari, Chaabane, Loukil, & Trentesaux, 2011)			•	•			•					•			•	•			Long jobs make it difficult to find a good solution	Tabu search and simulation-optimization obtained better results than the dispatching rules
(López & Arango, 2015)			•								•				•				User Interface	Algorithm adaptable to the user
(Lin & Chen, 2015)				•							•	•	•		•		•		Stochastic problem	Determine the optimal allocation of jobs from the production line
(Aurich, Nahhas, Reggelin, & Tolujew, 2016)			•	•						•	•	•			•	•			Jobs on time	Measure the effectiveness of the algorithms
(Yaurima, Tchernykh, Villalobos, & Salomon, 2018)	•		•		•						•				•			•	Energy consumption	Minimize time and energy consumption
(Ruiz et al. 2008)	•	•				•								•	•				HFSP skipping stages	Minimize makespan
(Yu, Semeraro, & Matta, 2018)	•				•						•					•			Dynamic programming	Improvement of time versus dispatching rules

Table 1. Literature Review

In order to know how the HFS problem has been solved, the keywords and the summary of the first 15 articles found in each search were examined. Furthermore, the remaining papers from the previous filter were read evaluating the production environment studied and the method approach to find out the solution. This was the last filter to include the research in this review. The information was divided into solutions based on exact methods, heuristics, metaheuristics and simulation-optimization as shown in Table 1. Literature Review.

Several authors have tried to solve this problem with exact methods algorithms, heuristics, metaheuristic and simulation-optimization models. According to the literature review chart, the first case aims to solve the HFSP considering simple scenarios (Carlier & Néron, 2000) using algorithms like Branch & Bound (B&B). Methods such as heuristics are used to reach an approximation of the solution, (Low, Hsu & Su 2008) using heuristics as dispatch rules. Metaheuristics propose general solutions for production problems, for this reason, (Fakhrzad & Heydari 2008) employs Taboo search (TS), (Zandie, Fatemi & Diattarr, 2006) use artificial immune systems or AI (AIS) and (Aurich, Nahhas, Reggelin, & Tolujew, 2016) give solution to the HFSP with a genetic algorithm (GA). Finally, (Lin & Chen 2015) and (Chaari, Chaabane, Loukil, & Trentesaux, 2011) use optimization simulation to resolve the situation raised.

2.2. Exact methods

The Exact methods are algorithms that find the optimal solution to a given problem. This technique uses sequentially steps to transform input variables to a set of output variables. In addition, such algorithms does not generate random numbers to be executed (Chieh, van Roermund, and Leenaerts 2005). The Branch & Bound algorithm (B&B) is one of the exact methods most used for troubleshooting scheduling. This method allows to find an optimal solution for the problems of programming in environments, because, it can be applied to a great variety of combinatorial optimization problems (Carlier & Néron, 2000). The process that follows B&B to find a solution is modeled as the ramifications of a tree (Yaurima, Tchernykh, Villalobos, & Salomon, 2018). Typically, each node in the tree corresponds to a sub-problem, defined by a task sub-sequence. Finally, to avoid enumerating all task permutations, the lower limit of the target functions is calculated at each step of each partial schedule (Carlier & Néron, 2000).

2.3. Heuristics

The Heuristics are algorithms based on intuitive ideas or useful techniques applied to search for good solutions to an optimization problem. These algorithms partially use randomization to search the solution, and usually involves iterative improvement techniques or rules. Even the method looks for good solutions, there is no way of finding out how close the chosen solution is to the global optimum (Chieh, van Roermund, and Leenaerts 2005). Often, the heuristics are designed for specific problems so that a “procedure that works for one problem cannot be used to solve a different one” (El-Sherbeny 2010). (Fakhrzad & Heydari, 2008) developed a heuristic based on three main algorithms. The main objective was to minimize the costs of late and early work, assuming the machines are identical at the stations. The first algorithm seeks to assign the work to the machines, converting the HFS problem to a Flow Shop (Fakhrzad & Heydari, 2008). The second algorithm schedules and sorts the jobs assigned to each machine by an approximation between the rules of dispatch EDD and JIT. Finally, in the third algorithm the resources for each job are leveled and the results of the proposed heuristics are compared with two metaheuristics: Simulated Annealing (SA) and Tabu Search (TS). (Fakhrzad & Heydari, 2008), based on their heuristics, demonstrated the ability to obtain superior results on the SA and TS metaheuristics.

Moreover, (Low, Hsu, & Su, 2008) propose another heuristic based on a model of n works organized in m classes or groups. There are two machine centers or stations where the work is processed. In the first station are m different machines, and in the second center there’s a single machine. The aim of the research is to minimize makespan. To achieve that, they develop a sequence assignment phase and a machine assignment phase. For the sequencing phase, consider four rules: (1) A random method, (2) SPT rule in the first season, (3) LPT rule in the second station, (4) Modification of the Johnson rule (Low et al., 2008). Later, it combines these 4 rules of sequencing with 4 rules of dispatch, to find an optimal solution or close to the optimal one. 16 possible combinations were generated to solve the problem in a computational way. It was concluded that the modification of the Johnson rule allowed the obtaining of better results (Low et al., 2008).

2.2. Metaheuristics

The optimal solutions are not always easy to obtain, even more when the complexity and the problem size gets higher. Metaheuristics are particular optimization techniques that can provide an acceptable solution to the optimization problem in a reasonable amount of time (Eskandarpour, Ouelhadj, and Fletcher, 2019). Unlike the exact methods, metaheuristics does not guarantee optimality. These algorithms work as “iterative strategies that modify heuristics by combining different moves for exploring and exploiting the search space”(El-Sherbeny 2010). Many investigations based on the problem of scheduling, do not take into account the dependence of the times of enlistment of the machines with the sequencing of the works. (Zandieh, Fatemi, & Diattar, 2006), under an immune algorithm (IA), developed a sequence of production taking into account enlistment times. The method of solution proposed (IA) allowed to reduce the dominance of good solutions to give priority to others and to find the best solution. Making a comparison between IA and GA, it is concluded that the effectiveness of the first is much greater since the second converges quickly. However, the solution generated by the IA algorithm, provides a shorter end time of the last task or Makespan.

Genetic algorithms are tools that have been used very often in solving the HFS programming problem. An approximation at the industrial level in the manufacture of tortillas was raised by (Yaurima et al., 2018). In this work, the objective is based on improving the end time of the last work and minimizing the energy consumption of the machines. The conditions of the production environment in this research involve 6 stations with 6 machines in each. In addition, the work is grouped according to the type of tortilla that is elaborated. The compatibility of the work is taken into account to be processed simultaneously. That is to say that the works belonging to the same group can be processed at the same time. In addition, the proposed case takes into account the time of set-up and buffers. The author gives solution to this problem by means of a genetic algorithm (GA), reducing in more than 48% the production time and in 47% the energy consumption.

Yu, Mamerao & Matta (2018) used of a genetic algorithm to generate production scheduling with unrelated machines and eligibility restrictions. In addition, Yu, Cameraman & Matta (2018) developed a new method of decoding based on dynamic programming. It was obtained the improvement of problems of re-sequencing and control of dead times.

Finally, Lopez & Arango (2015) developed a genetic algorithm (GA) for an HFS environment. The algorithm included factors close to the reality of the industries. Research, gender models and programming scenarios from five variables: job numbers, population size, iteration number, mutation rate and crossover points. In addition, the genetic algorithm proposed genre computational times of interest to the business sector, lasting less than a minute in its execution.

2.3. Simulation-Optimization

The real-world systems can be optimized combining heuristic optimization methods and simulation based modeled (Ferrer, López-Ibáñez, and Alba 2019). The computer simulation procedure can be used in evaluating complex systems. Thus, simulation-optimization provides a method to establish an optimal value of the decision variables in the system, where optimal is measured by a function of output variables generated by the simulation model (Swisher et al. n.d.). The simulation-optimization is capable of capturing interactions between different entities in a complex system in order to identify it better. (Lin & Chen, 2015) developed a sequence of works in the manufacture of semiconductors using this tool. Currently, the process associated with the production of this type of products requires multiple stages and depends on the customer's request. As a result of the above statement, the machines to be used are not the same to produce semiconductors. (Lin & Chen,2015), use metaheuristics and acceleration techniques such as genetic algorithm (GA) and optimal budget allocation or optimal computing budget allocation (OCBA) to give solution to the problem posed. As a result of the investigation, the authors gave solution to the problem of assignment of the works for each one of the orders, minimizing the flow time or flow time (F).

A printed circuit (PCB) is a plate made from an insulating material that contains copper pathways that interconnect the components of a circuit. The production environment where these elements are made is HFS. (Aurich, Nahhas, Reggelin, & Tolujew, 2016), develops an approximation to give an optimal solution to the PCB scheduling. With the objective of minimizing the final time of the last task and the total delay, the authors use simulation based on optimization algorithms (ISBO), simulated annealing (SA) and Tabu search (TS). The investigation allowed to conclude that simulation-optimization finds a solution much faster than SA and TS. However, the solutions provided by the two metaheuristics generated better results than those supplied by ISBO.

This literary review showed the little research directed towards the method of simulation-optimization solution. In addition, methods such as heuristics and metaheuristics have been related in greater proportion to the problem of programming production in HFS environments. The use of the simulation-optimization method would be beneficial for this project, due to its ability to approach an optimal solution under the analysis of multiple possibly real scenarios.

3. Objectives

***General Objective*.**

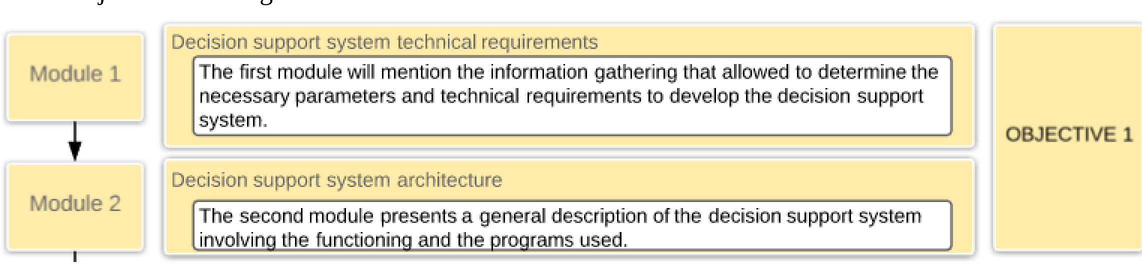
Develop a decision support system that provides production scheduling in the chemical area of the company Fuller Pinto, based on Simulation-Optimization model and minimizing the total weighted tardiness.

Specific Objectives:

1. Establish the requirements of a decision support system for the scheduling of Fuller pinto at technical and computational time level.
2. Build a scheduling model based on a simulation-optimization approach, according to the operation characteristics of chemical products area of Fuller Pinto.
3. Validate the simulation-optimization model through random instances and compared them against dispatching rules and backtesting.
4. Consolidate the decision support system through a user interface to solve the production scheduling of detergents.
5. Evaluate the economic impact of the proposal, by comparing the solutions obtained by the current method used by the company and the proposed model.

4. Methodology

This section explains the methodological steps to create the decision support system for scheduling the production of Fuller Pinto. In order to track the development process, each step will correspond to a *Module* that contain the information and explanation of each phase. The explanation modules and the corresponding specific objective to the right are described below:



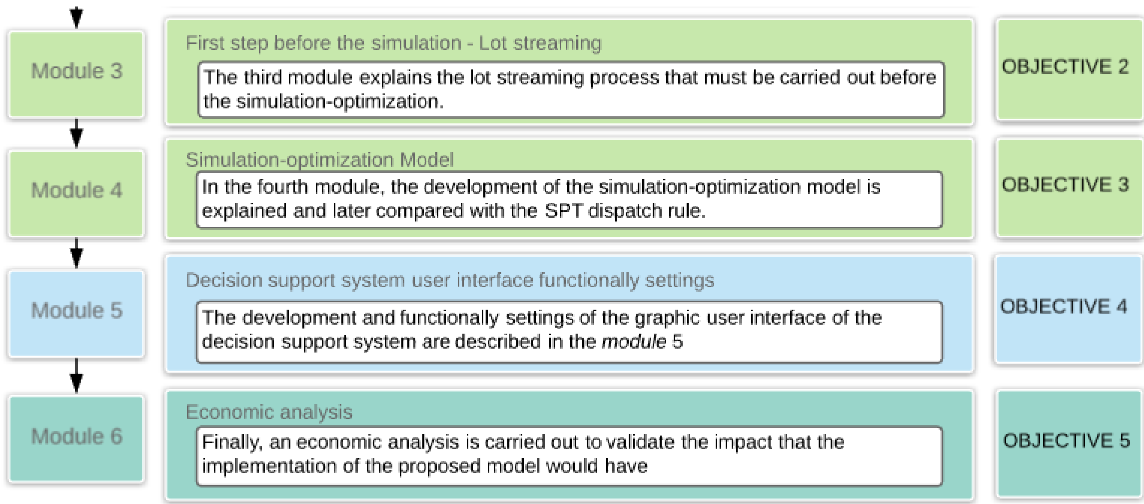


Figure 4. Methodological Steps for the Decision Support System

4.1. Module 1: Decision support system technical requirements

The first step to develop the decision support system is to know the production manager of Fuller Pinto technical and computational requirements. In order to meet these requirements, the Quality Function Deployment (QFD) is going to be performed. The main goal of QFD is to know consumer needs and then translate the consumer request into design (Maynard and Zandin 2001). Getting deep into the customer demands could guarantee major quality products. In addition, it can be considered the QFD results as the authentic consumer's voice, since it brings his voice and wishes. First, the list of requirements was designed together with the production manager considering the current production environment. After the checks and the approval of the production manager a survey was made to know the importance of the final requirements. The survey is already attached in the Appendix-Table 1. The final requirements are classified according to the user manipulation, operation and results (See Table 2). The requirements represent the consumer voice in the Quality Function Deployment. The requirements description is presented on Table 2.

Customer requirements classifications		
User Manipulation	**Operation**	**Results**
Accessibility	Ease of use	Graphic treatment of the results
The decision-support system can be used by any user or work employee authorized.	The design must allow the manipulation to be practical, simple and effective	The visual design of the decision support system must present the results in an orderly, pleasant and synthesized manner
Ease of information entry	Ease of installation	Safety
The decision support system is expected to be able to receive the full data order required for a campaign	The decision support system must be supported and installed for a device with windows operating system. The installation process should be for a user with basic computer skills	Ensure that the results are properly saved to prevent loss
	Capacity to adapt to the environment	Final result
	It must reflect the production dynamic of the company (campaigns, products, processes, times, machines)	The decision support system is expected to present an optimal sequence for the production of the company's work
	Run time	
	Should not be more than two hours	
	Simultaneity	
	The decision-support system should not block the simultaneous use of programs	

Table 2. Classification of customer requirements

Once customer requirements are ready, the design features and technical tools that are going to be aligned to accomplish the customer requirements are set up. These features represent the techniques that answer the question "¿How the designed system going to solve the consumer demands?". In order to resolve this question, the ISO/IEC 25010 standard for software and applications design is going to be used. This standard classifies the software design requirements in the general guidelines described below.

Figure 5. Decision Support System Technical Features identifies the characteristic and sub-characteristic of the software design standard. Both of them represent the quality model categories and sub-categories defined in ISO/IEC 25010, and each sub-category represent the technical feature that the decision support system designed will provide (International Organization for Standardization 2011).

Decision support system technical features		
Functional Suitability ***Sub-characteristics*** **FC:** Functional Completeness **FCr:** Functional Correctness **FA:** Functional Appropriateness	FC	The decision support system considers all the products of the company and covers all the tasks and objectives of the user.
	FCr	The decision support system will seek results with a high level of precision using a local search algorithm
	FA	The decision support system will have a set of functions in order to help the user to accomplish his tasks.
Compatibility ***Sub-characteristics*** **CO:** Coexistence **IN:** Interoperability	CO	The application is capable to coexist with other programs.
	IN	The decision support system will be able to exchange information between the following software's: Visual Basic from Excel, Java, Netlogo and Python.
Reliability ***Sub-characteristics*** **MA:** Maturity **AV:** Availability	MA	The decision support system is reliable and can complete the functions for which it was designed
	AV	The decision support system will be available for use at the desired time
Security ***Sub-characteristics*** **CON:** Confidentiality **INT:** Integrity	CON	The decision support system offers to protect the company's data and information.
	INT	Capacity of the system to prevent access or unauthorized modifications to data or computer programs.
Usability ***Sub-characteristics*** **AR:** Appropriateness Recognizability **LE:** Learnability **OP:** Operability **UEP:** User Error Protection **UIA:** User Interface Aesthetics **ACC:** Accessibility	AR	The decision support system allows the user to understand if it approaches his needs. This means offering the optimal sequencing of the jobs entered by the user.
	LE	Refers to the ability of the decision support system to be learned by the user
	OP	The decision support system offers to the user a simple and intuitive manipulation
	UEP	The decision support system offers an interface capable of correct errors when entering the information.
	UIA	The decision support system offers visually pleasing graphical interaction.
	ACC	The decision support system will be used by different users
Performance Efficiency ***Sub-characteristics*** **TB:** Temporal Behavior	TB	The decision support system response and processing time will be according to the functions and conditions of the background process required for the client.
Portability ***Sub-characteristics*** **AD:** Adaptability **INS:** Installability	AD	The decision support system can be installed and executed in different PC's
	INS	The programs used for the decision support system can easily be installed an be downloaded for free.

Figure 5. Decision Support System Technical Features

4.1.1. Quality Function Deployment results

Once the customer requirements and technical features are clear, the QFD matrix can be done. To achieve that, the customer requirements are arranged on the matrix rows while the technical features are on the top. In addition, the *"rooftop"* of the QFD is included above the technical features resulting in the figure shown below.

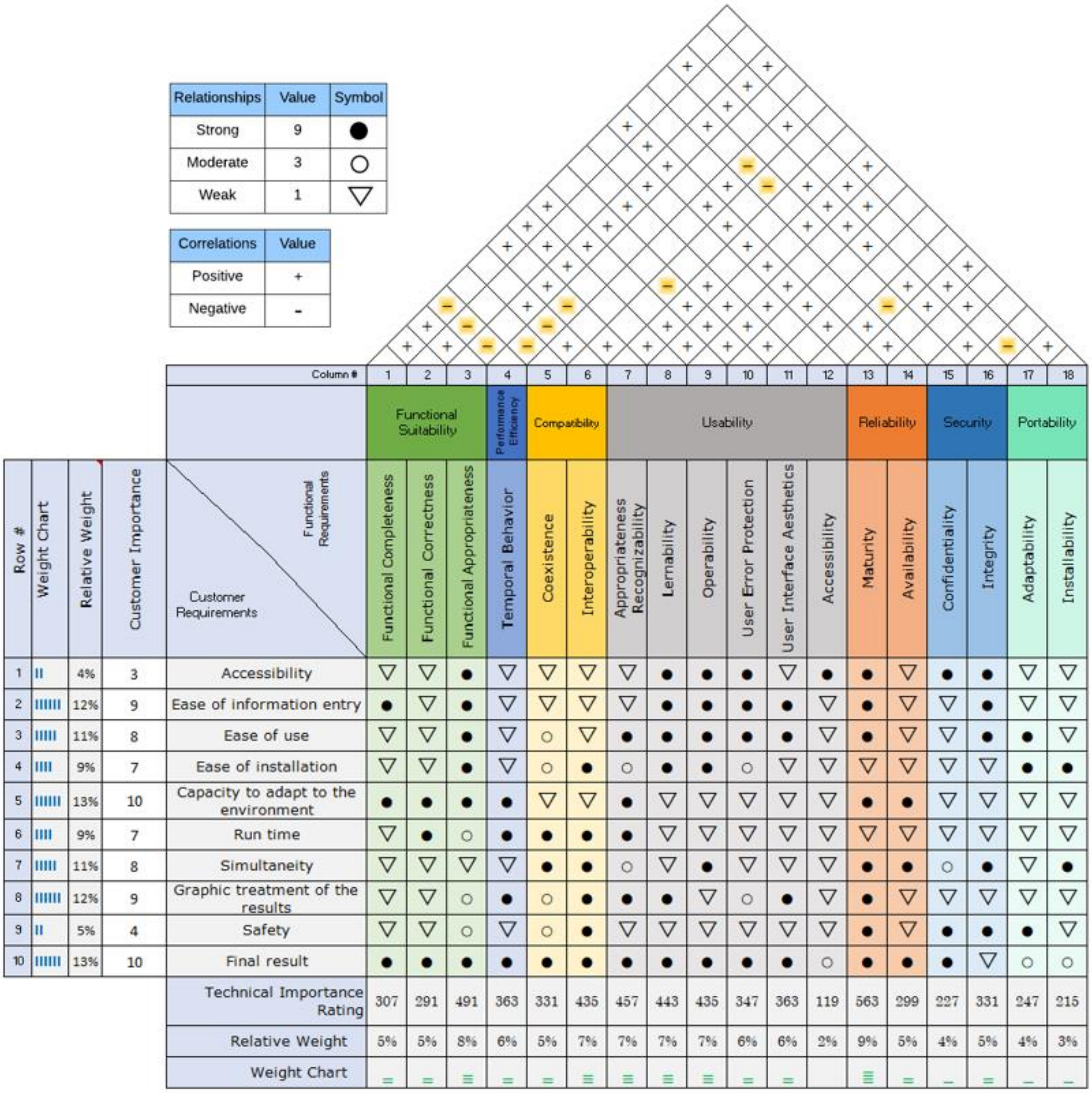


Figure 6. Fuller Pinto QFD

Figure 6 summarizes each consumer requirement importance value to the left. It also shows the matrix relationship symbols indicating the strength of the relation between the technical and the consumer requirements according to the scale. The QFD *"rooftop"* refers to the correlation matrix that evaluates the relationships among the technical requirements considered in the Figure 6. Fuller Pinto QFD scale.

Once the QFD is completed the customer requirements are translated into technical requirements or design specifications for the decision support system (Maynard and Zandin 2001). At this point, the QFD matrix has enough data to select the key technical requirements to further focus on. The importance weighting is calculated by multiplying the customer importance value with the value assigned to a relationship in the matrix (Maynard and Zandin 2001). Then the technical feature column is totalized to find out its importance weight. Although none of the percentages of the technical requirements stands out significantly over the others, without a doubt the most important is the requirement of Maturity. This requirement is followed by Functional Appropriateness feature and the group of Appropriateness Recognizability, Learnability, Operability and Interoperability. This last group of features match with 7% of weight. Similarly, the technical requirement that is not critical to achieve the customer requirements is the Availability.

The previous results are extremely useful in the decision support system design process. Once the most important technical features have been detected, the decision support system development will be focused primarily in the development of technical elements that guarantee four main points:

- ✓ A system that satisfies the execution of user tasks while the system is being used.
- ✓ A system able to be learned and simply manipulated in an easy and intuitive way.
- ✓ A system that provides the complete information required for the production scheduling process.
- ✓ A system that achieve its principal objective, providing the scheduling for the chemical area of Fuller Pinto.

Regarding to the *rooftop* of the QFD a qualitative analysis can be performed. Taking advantage of the technical requirements found recently a correlation analysis can be performed. Considering that the decision support system design is going to be focused in some of the technical features that have higher importance, it must be analyzed how the other parameters are going to be affected. In conclusion, it can be useful to know how much a parameter could be pushed at the cost of the other.

The best way to do this analysis is to compare each pair of technical requirements and identify if they have a positive or negative correlation. If they have a positive correlation between each other, that means that the increase of one requirement will affect the other on a positive way. On the other hand, if the correlation is negative that means that an increase in one of the requirements will affect negatively the other. The Temporal behavior requirement is a good illustration of a parameter with negative correlations. As far as it is known, the Appropriateness Recognizability, the Functional Appropriateness and the Temporal behavior requirements are some of the most important features. Even though, the two first requirements mentioned have a negative correlation with the Temporal behavior. This implies that although they are of similar importance, focusing the design on functional aspects that allow the user to execute all the required tasks and obtain the optimal schedule, affects negatively the response time since the development of these functionalities causes the system to become heavier at computational level.

Another negative relationship is presented between requirements such as Coexistence and Confidentiality. In this case, the decision support system platform that is going to be designed is actually able to coexist with other programs used in the company and allows the user to use them simultaneously. However, the decision support system does not guarantee that a background process of those programs could access or manipulate the company’s data or information.

The previous results of key technical requirements to satisfy the main needs of the client are extremely useful for the design of the decision support system. As a consequence, the software development efforts have been focused on implementing the appropriate programming functions and characteristics to guarantee a reliable system that presents the jobs schedule of the chemical area. To see the previous results reflected in the software developed and also describe its internal operation, the description of the architecture of the decision support system is presented below.

4.2. Module 2: Decision support system architecture

The decision support system works as an application that uses different software’s to get the scheduling of the different jobs included on a campaign. To describe the interaction between software’s the Figure 7 is shown. In this figure a couple of boxes with a top number refer to the software’s that are involved in the decision support system. Between two boxes a double rectangle could appear. This means that a CSV file will be used to send information from the program to the left and receive information to the program to the right.

Therefore, the decision support system starts with the Python user interface that receive the input data. This information is set on a CSV file sent to the Visual Basic program in Excel. The process made in Visual Basic from Excel generates an output file received in Java. Java libraries are connected directly to NetLogo, which is the software that will run the simulation. When the simulation has finished a CSV file is again generated to design a Gantt Chart with Python.

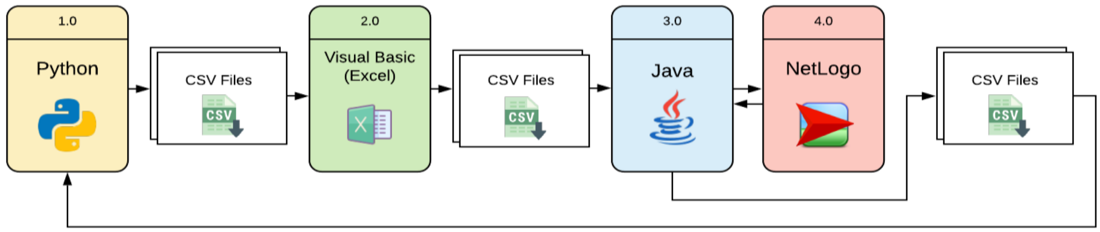


Figure 7. Architecture diagram

On the other hand, another diagram will be used to explain how the information goes throughout the decision support system. The Unified Modeling Language (UML) is an international standard frequently used to design sketch, diagrams and develop documentation related to software development (Barclay and Savage 2004) . The main objective of UML Diagram is visualizing a software program using a collection of diagrams (Barclay and Savage 2004). The current UML standards identify 13 different types of diagrams classified in two global groups: Structural UML Diagrams and Behavioral UML Diagrams. In order to describe interactions and the exchange of information among the different programs used in the decision support system, the UML Sequence - Interaction Diagram is going to be used(Barclay and Savage 2004). The diagram is shown in Figure 8. UML sequence-interaction diagram.

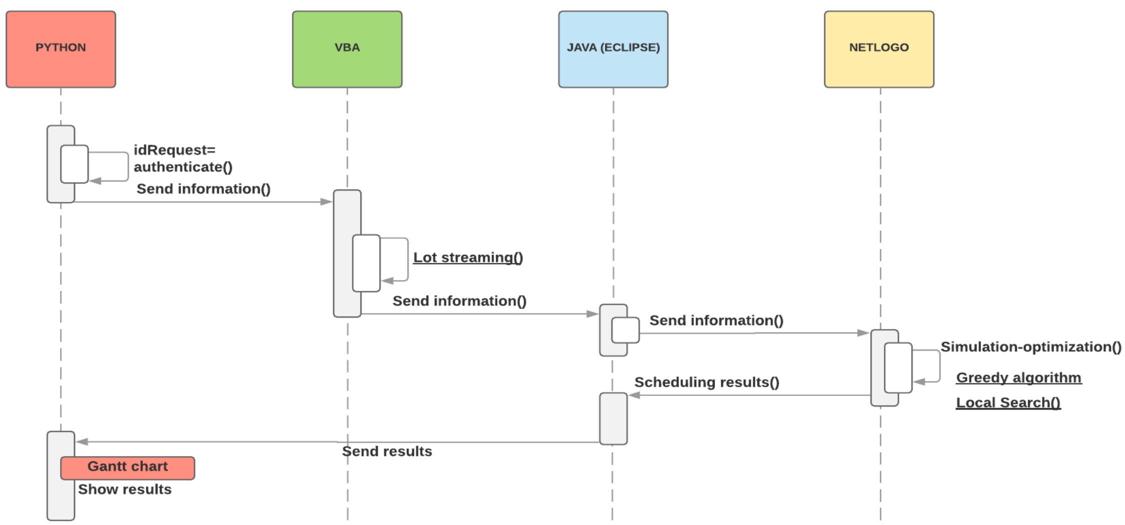


Figure 8. UML sequence-interaction diagram

Figure 8 synthesizes the UML diagram reflecting the different software in the system and the information shared between each other. The top boxes represent the decision support system actors. The principal actor is User Interface made with programming language Python. This interface allows the user to manipulate the decision support system and will display the final result. In addition, the interface allows the user to enter all the data information of products, quantities and due dates that are registered and sent to VBA. Then, the VBA program interprets the information of product quantities and generates the batch partitioning known as lot streaming. Later the Java code receives the CSV file with the batch partition and prepare that information to run the simulation-optimization model in NetLogo. It is important to highlight that during the execution of the decision support system, the VBA process, and the simulation model called and run from Java will not be displayed on the screen. Finally, the Java code catch the output data generated by NetLogo and send it back to Python user interface to display the Gantt Chart automatically.

To guarantee that the screen will only display the graphic user interface, the VBA and Java procedures have been turned into a *VBS* and *jar* files respectively. This format files allows the system to run those procedures on background and looks for the system to be friendly according to the graphic requirements found previously in QFD analysis.

4.3. Module 3: First step before the simulation – Lot streaming

Before getting deeper into the simulation model a batch partitioning method must be developed. This section presents the lot streaming method for the jobs ordered. As it was stablished in the objectives, the main purpose of this project is to schedule the jobs of chemical area of Fuller Pinto due to the fact that exist tanks capacity restrictions. Nevertheless, the schedule cannot be done properly if the amount of mixture for each tank is not determined previously. This problem is known in the literature as lot streaming. “Lot streaming is a process of breaking a batch of jobs into smaller lots, and then processing these in an overlapping fashion on the machines. This important concept can significantly improve the overall performance of a production process, and thereby make the operation of a manufacturing system lean” (Sarin & Jaiprakash 2007).

According to the previous statement, it is necessary to develop the batch division step before the simulation because the output data of this process serve as the input for the simulation. Since the first step before starting any production run is to enter all orders, it is important to note that only the number of units to be processed is known. However, the first stage in the shop-floor needs the number of liters to mix on a tank, not the numbers of units entered before. Therefore, the total number of liters of each product ordered is going to be divided into different batches using the following methodologies. Finally, each methodology is evaluated to set the best one into the decision support system. It is also important to emphasize that none of the methods seeks to optimize the lot streaming. As it was said before, the aim of this research is to generate the scheduling plan, and the optimization of the batch partitioning process will not be considered. For this reason, the methods seek a good and a feasible solution. So, the scope of the batch division will be made only for the mixing stage. Also, a batch will not have the same amount in the sublots divided, on the contrary, it will vary according to the capacity of the tank. Finally, it is important to highlight that lot streaming methods and later in the simulation-optimization process, only deterministic events were considered. No random event is taken into account in this model.

Three methods were developed for the problem. The evaluation was made based on three performance indicators: the computing time, the waste in the tanks and the number of tanks used. Besides, there are policies on the mixing stage that restricts the problem and must be considered by the methods.

Manufacturing Policies for the chemical area

- The current capacity of the plant is 23 tanks, differentiated into three types: 6 tanks of 16,000 liters, 2 tanks of 8,000 liters, 12 tanks of 4,000 liters, 1 tank of 1,000 liters and 2 tanks of 400 liters. These last 3 tanks of smaller capacity are exclusively for special preparation products.
- It is important to clarify that the tanks are not used at their maximum capacity, but up to 500 liters less than their maximum limit.
- Tanks of 16,000 liters: They will only be used for sublots with amounts to mix between 8000 and 15500 liters.
- Tanks of 8,000 liters: They will only be used for sublots with amounts to be mixed between 4000 and 7500 liters.
- Tanks of 4,000 liters: They will only be used for sublots with amounts to be mixed between 500 and 3500 liters.

The input data of this methods are the name of the product, units to produce of each product, volume per unit, product status and the campaign to which it belongs. The output data are the name of the product, units produced of each product, volume per unit, initial tank assigned, machine assigned, product type, product status, initial release order and the campaign. As evidenced in the UML diagram, the input and output data of the model are type CSV. The input data comes from Python, while the output data will be used by Java. Table 3 shows the pseudocodes of each method considered for the lot streaming problem. The function of the first method is to fill a tank at a time randomly taking into account the production requirement until a tank fill the missing amount. On the other hand, method two is not random, it fills tanks at a time respecting their maximum capacities a. Meanwhile, the third method is similar but not equal to the first method.

The purpose is to evaluate the best alternative for company requirements and the designed support system. The normality assumption was first evaluated in the *Statgraphics* software. However, the Appendix-Table 2, Appendix-Table 4 and Appendix-Table 6 determine that with a p-value of 0.0002, 0.0002 and 0.0006 for the performance indicators waste, computational time and number of tanks respectively, the samples do not have fit to a normal distribution.

Therefore, through the *SPSS* software, the evaluation was carried out by a non-parametric test, Kruskal-Wallis. The test was made for each performance indicator. The experiment evaluated each performance indicator with the three methods in batches of 5, 10, 15 and 20 products. Each one of the treatments were replicated four times. According to the test, a significant difference between the methods exists for each performance indicator. Also, the "Batch" factor evidences an important difference that represents a considerable variability as the batch quantity gets higher. The pairwise comparison of methods established that the second method response better to the requirements and restrictions for the problem. The test shows that with a p-value less than 0.05 for each indicator, the distribution of waste, computational time and number of tanks, is different across categories of method and batch. The Appendix-Table 3, Appendix-Table 5 and Appendix-Table 7 have the results of the non-parametric experiment for each performance indicator.

Pseudocode Method 1	Pseudocode Method 2	Pseudocode Method 3
Step 0: Initialization. Set *n* the total number of products in campaign *c*. Each *p* product will be related with the variable *acum[p]* which will start at 0 for each *p* product. Also, each p product will have a parameter called *requirement[p]*. This parameter contains the total amount required to be produced of each product. **Step 1: Random filling.** The *requirement[p]* will be divided in tanks. A tank t is created for product *p*. The *quantity[p,t]* corresponds to a fraction of the product requirement and is set randomly between 0 and 14500 liter for each new *t* created. **Step 2: Accumulation.** For each new tank created, the acum[p] variable will save the fractions of requirement[p] ($acum(p) = \sum_{t \epsilon T} quantity(p,t)$) **Step 3: Insertion.** If *acum[p] < requirement[p]* then create a new tank and fill the variable quantity[p,t] aleatory such as in Step 1 and go to Step 2. If *acum[p] > requirement[p]* then clear the value of quantity[p,t] of the last tank created and put the missing amount. On the contrary repeat this Step. **Step 4: Increment.** When *acum[p]=requirement[p]*, increment *p* to *p+1* until *p=n* and return to step 0.	**Step 0: Initialization.** Set *n* the total number of products in campaign *c*. Each p product will be related with the variable *acum[p]* which will start at 0 for each *p* product. Also, each *p* product will have a parameter called *requirement[p]*. This parameter contains the total amount required to be produced of each product. **Step 1: Restricted filling.** The *requirement[p]* will be divided in tanks. A tank t is created for product *p*. The *quantity[p,t]* corresponds to a fraction of the product requirement. The value of the quantity parameter is set respecting these criteria: [1] If requirement[p] $\geq$ 15500, then *quantity[p,t] = 15500* [2] If requirement[p] $\geq$ 7500 and requirement[p]< 15500, then *quantity[p,t] = 7500* [3]] If requirement[p] $\geq$ 3500 and requirement[p]< 7500, then *quantity[p,t] = 3500* The missing amount, r_p, must be assigned in one or more tanks depending on the requirement and respecting the same criteria. Also, if this value does not respect the restrictions [1],[2] and [3], the variable *quantity[p,t*] takes the same value of the r_p. **Step 2: Accumulation.** For each new tank created, the acum[p] variable will save the fractions of requirement[p] ($acum[p] = \sum_{t \epsilon T} quantity[p,t]$) **Step 3: Missing amount.** If *acum[p] < requirement[p]* then $r_p =$ *requirement[p] – acum[p].* **Step 4: Increment.** When *acum[p]=requirement[p]*, increment *p* to *p+1* until *p=n* and return to step 0.	**Step 0: Initialization.** Set *n* the total number of products in campaign *c*. Each *p* product will be related with the variable *acum[p]* which will start at 0 for each p product. Also, each p product will have a parameter called *requirement[p]*. This parameter contains the total amount required to be produced of each product. **Step 1: Random filling.** The *requirement[p]* will be divided in tanks. A tank t is created for product *p*. The *quantity[p,t]* corresponds to a fraction of the product requirement and is set randomly between 0 and 14500 liter for each new *t* created. **Step 2: Accumulation.** For each new tank created, the acum[p] variable will save the fractions of requirement[p] ($acum(p) = \sum_{t \epsilon T} quantity(p,t)$). **Step 3: Insertion.** If *acum[p] < requirement[p]* then create a new tank and fill the variable quantity[p,t] aleatory such as in Step 1 and go to Step 2. If *acum[p] > requirement[p]* then eliminate all the given solution and return to Step 0 until the model find a solution where *acum[p]=requirement[p].* **Step 4: Increment.** When *acum[p]=requirement[p]*, increment *p* to *p+1* until *p=n* and return to step 0.

Table 3. Pseudocode Methods

4.4. Module 4: Simulation-optimization Model

A simulation method can be useful to evaluate the performance of a certain schedule on a HFSP. In this case, the simulation tool selected was an agent-based simulation software called Netlogo. This simulator is open source, and it can illustrate the structure and behavior of complex system. The agent-based simulator was selected among other software because its proficiency representing the batches as agents that changes their status in time. It also can simulate the batch behavior while it interacts with other objects or agents in the simulation. As previously stated, the production environment of the chemical area of Fuller Pinto has clearly two stages with parallel machines on each stage and eligibility machine option. Certainly, restricted to lot size and machine capacity. In terms of the manufacturing environment notation this scenario is described as:

$$FF2, (RM23)^1, (RM6)^2 \mid M_j, S_{jk} \mid \sum_j \mathrm{W}_j T_j$$

Equation 1

The manufacturing environment notation presented above shows how the shop-floor is organized, the constraints and the objective function considered ($\sum_j W_j \times T_j$) . In this context the shop-floor is Hybrid Flow Shop or Flexible Flow Shop with two stages ($FF2$) that has 23 unrelated machines (RM23) in the first stage and 6 unrelated machines (RM6) on the second stage. In addition, both stages are subject to the restrictions of machine eligibility (M_j) and sequence depend setup times ($S_{j,k}$) evaluating the total weighted tardiness. As a result, the Hybrid Flow Shop environment describe properly the chemical area operations.

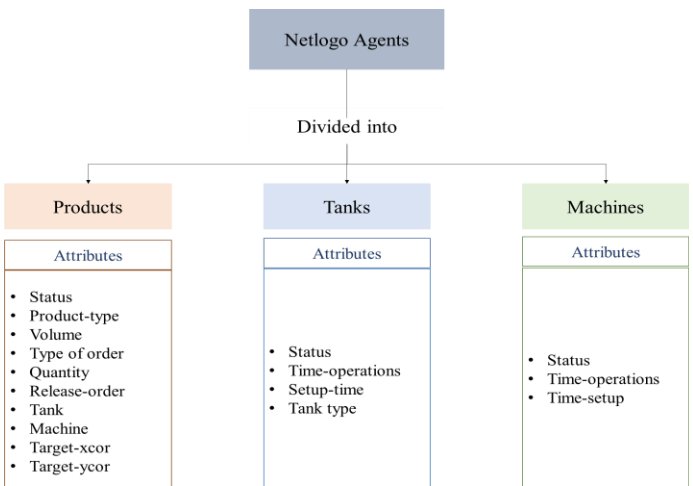


Figure 9. Netlogo Agents

In this context, it must be highlighted the important proposal of the lot streaming procedure done previously. At this point, the output variables of the batch division are going to be used as the input data to stablish and create the simulation model agents and features. Each batch coming from the lot streaming is named in Netlogo as a product and every attribute of them comes from it except for Status, Target-xcor and Target-ycor. Furthermore, two more agents were created in order to simulate the production process of the chemical department of Fuller Pinto's company. Figure 9 demonstrates the agents and features created.

The attributes of each agent can lead to affect or disturb the system by having influence on the processing and transport decisions. Table 4 shows the products attributes.

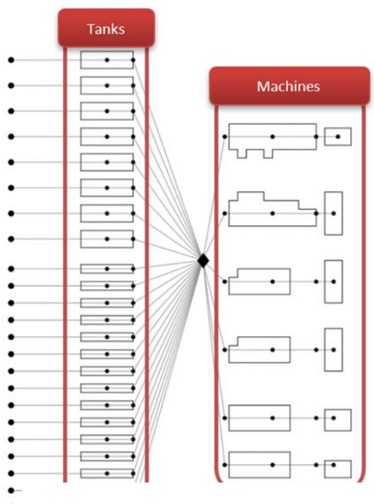
Products attributes		
Status	**Release-order**	**Product-type**
The status attribute allows to know if the batch is "on movement", "stopped", "finished" or processing "in tank" or "in machine".	This attribute gives the sequence order to be released.	The chemical area works with two types of products, viscous or liquids
Tank	**Volume**	**Machine**
Indicates where the agent must be mixed.	Refers to the bottle presentation (ml).	Indicates where the agent must be processed.
Type of order	**Target x-cor**	**Target y-cor**
Refers to the batch partitioning. For instance, if a job has been divided in three batches or groups, there will be three agents on the simulation. The first batch will have the 'batch 1' label, the second	This attribute helps the agent to save the coordinate of X axis on the simulation screen	This attribute helps the agent to save the coordinate of Y axis on the simulation screen

Table 4. Products attributes

On the other hand, tanks and machines have similar features. Table 5 describes the features.

Tanks and machines attributes			
Status	**Time-operation**	**Setup-time**	**Tank-type**
The status attribute allows to know if a tank or a machine is "mixing" or "processing" a batch respectively.	The time – operation attribute indicates the processing or mixing time that a batch must complete.	Refers to the time spent cleaning and preparing the tank or machine before a product is started.	Indicates the volumetric capacity of the tank (Liters)

Table 5. Tank and Machines Attributes



Once the agents and its attributes are defined the environment layout is designed. This sketch is created by coding the appropriate scripts in the *Setup* procedure of NetLogo. Figure 11 allows to see graphically machines, tanks and links between them.

When the layout sketch is ready, the simulation can be run. The execution can be done with the Go procedure. This procedure executes other subroutines. The first subroutine runs a Greedy algorithm that allocates the input batches into the tanks looking for maximize the utilization of each one of them. This optimization algorithm is known to be voracious, which implies that once a decision has been made, it is not reconsidered anymore. Once the Greedy procedure has finished, the subroutine that contains all the logic that simulate the production environment is executed.

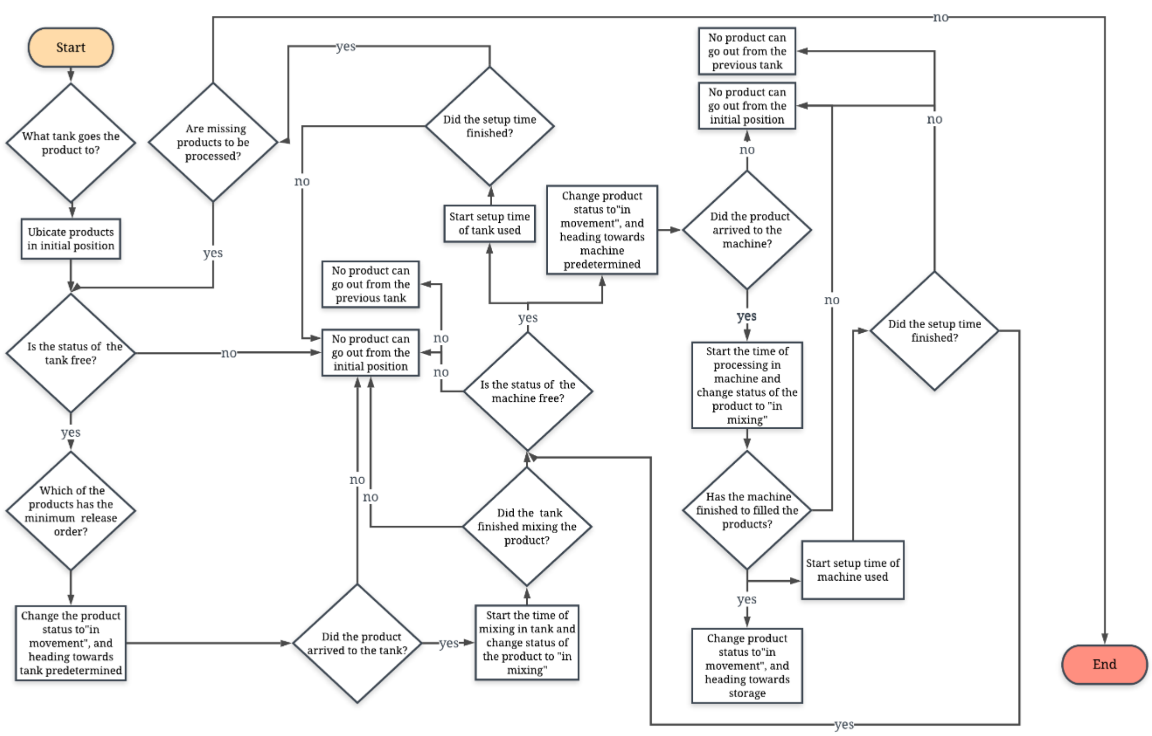


Figure 10. Product Route

Figure 10 shows the route of a product throughout the shop-floor follows the logic. Finally, a post-optimization process is performed in order to improve the objective function by making changes to the release-order variables and assignment to tanks of the batches.

The finals results of the Simulation-optimization model are:

i. <u>*Weighted tardiness*</u>:

This indicator is calculated by unifying all batches who comes from the same orders. The maximum finish time in machine between all batches from a given order is taken and will be called completion time of the job (ct_j). Also, the due date *(dd)* considered was 20 days, each day of 16 hours. Accordingly, the weighted tardiness was calculated by comparing the due date with the maximum finish time of an order. If the completion time of the job is higher than the due date, the weighted tardiness takes a positive value. On the contrary, it takes the value of zero (Figure 12).

If the weighted tardiness takes a positive value is because the difference between completion time of the job and the due date is greater than zero. This difference is called tardiness and is represented in Figure 10 as (t_j). On the other hand, the calculation of the weighted tardiness is a weighted sum which depends on the situation of the job. The situation is divided into two options: the order is delayed or is on time. “Delayed” orders come from previous campaigns to the one in current process and, “on time” are the orders which belong to the current campaign. By virtue of the foregoing, the weight of “Delayed” orders is 3000 and of “on time” orders are 1000. The main reason of the previous statement is that delayed orders have 3 times more importance than on time orders.

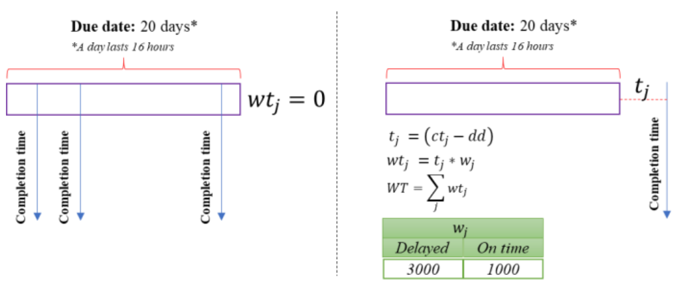


Figure 12. Decision Making Process Through the Shop-Floor

In addition, the model calculates the weighted tardiness when the due date is decreased by 15%, 25% and 35% with the purpose of comparing it with the performance of a dispatching rule. The performance will be evaluated by an experiment design described later. When the weighted tardiness of each job (wt_j) is calculated, the global weighted tardiness is given by the sum of them *(WT).*

ii. <u>*Supply percentage*</u>

In consequence of the job schedule given by the simulation-optimization model exist the supply percentage. This indicator shows what percentage of orders were accomplished in a campaign. First, it is calculated a supply percentage per job. Then an average between all the percentage results of all the jobs is calculated to obtain the performance of compliance of orders by campaign. This percentage is calculated before the due date is reached. After that, if a job was not made, the supply percentage will be considered as 0%.

iii. <u>*Job scheduling*</u>

As a result of the simulation-optimization process the main data of the execution is recorded and saved into a CSV file. The data acquisition includes the variables details that must be used to design the Gantt Chart that shows the result scheduling. Therefore, the required information is the product name, the tank and machine that the product have visited, and finally the start and end time in both the tank and the machine.

4.4.1. Greedy

The greedy algorithm is proposed to balance the use of tanks and avoid situations where exist a large number of waiting product to be mixed. This algorithm gives initial values for the next algorithm to be executed. However, it cannot get optimal solution and its results give a local optimal solution (Wang et al. 2018). It is important to mention that the Greedy algorithm respect the tanks capability constraints. The input of this algorithm is the output from the lot streaming which indicates if a product can be processed in a 16000 liters tank or in a tank of 400 liters.

Products are assigned to tanks in a sequential order. First, they are divided in five categories. Table 6 illustrates each category represented by every type of tanks that exist. Afterwards, they are distributed starting in tank one of each category. When a product is put in the last tank of a category, the next is allocated again in the first one and so on.

Category	**1**	**2**	**3**	**4**	**5**
Elements	Products that must be mixed in tanks of 16000 liters	Products that must be mixed in tanks of 8000 liters	Products that must be mixed in tanks of 4000 liters	Products that must be mixed in tanks of 1000 liters	Products that must be mixed in tanks of 400 liters
Number of tanks per category	●●● ●●●	●●	●●●●●● ●●●●●●	●	●●

Table 6. Product Assigment Categories

In order to show the previous process, Table 7 presents the following pseudocode.

<u>Greedy algorithm</u>

Step 0: Initialization. Distribute the *p* products in the category they belong to.
Step 1: Movement. Assign one by one *p* product to each *m* tank in each *n* category sequentially. If a product is assigned to the last *m* tank of *n* category, the product must be allocated in the first tank of the category.
Step 2. Stopping. If all *p* products were assigned maximum to one tank.

Table 7. Greedy Algorithm Pseudocode

4.4.2. Post - Optimization

Once the simulation starts running, the Local Search algorithm is activated in order to compare the values of the performance indicator weighted tardiness between several iterations. This is an improvement algorithm that tries to obtain a better schedule by manipulating the current schedule. A local search procedure does not guarantee an optimal solution (Michael L. Pinedo 2016). According to Michael L. Pinedo (2016) the general criteria for local search algorithms are the following:

i. The schedule representation needed for the procedure.
ii. The neighborhood designs.
iii. The search process within the neighborhood.
iv. The acceptance-rejection criterion.

The proposed algorithm attempts to find a job schedule that is better than the current one in the neighborhood (Michael L. Pinedo 2016). Also, the local search script is a procedure located in NetLogo. This procedure controls all the simulation logic and simultaneously executes the search, accepting or rejecting a candidate solution. The local search has the capability of running the model a limited number of times (100 times) maintaining the results given by the greedy algorithm for every 5 iterations *(Diversification criteria)*. Furthermore, for every iteration it changes the release order attribute of products *(Intensification criteria)*. In addition, the algorithm must evaluate the weighted tardiness performance por each iteration and compare it between them to improve the solution. As a consequence, if the solution has not improved over 10 iterations, the algorithm will stop and give the best job schedule. In order to show the previous criteria in the proposed algorithm, Table 8 presents the following pseudocode.

Local-Search algorithm

Step 0: Initialization. Set the number maximum iterations $n = 100$ and the number of the stop criteria $m = 10$.
Step 1. Current solution. Wait until the first iteration has run and save the results of the weighted tardiness on an S variable in order to save the value of the performance indicator of the current solution.
Step 2. Process. Once a simulation has run, the algorithm must change aleatory the release order of each product. It cannot be two products with the same number of release order. Also, for every 5 runs, the greedy algorithm will give other solution of tank assignment.
Step 3. Stop criteria. When the n iteration is finished, compare the current solution S with the weighted tardiness result of the n iteration, wt_n. If the result of S is better ($S \leq wt_n$) , $X = X + 1$. On the contrary, if the wt_n is better than S ($S \geq wt_n$), S = wt_n and X starts in 0 again (X = 0). The Local search must stop when $X = m$ or when the simulation has run n times.

Table 8. Local Search algorithm pseudocode

4.4.3. Validation of the simulation-optimization model

4.4.3.1. Simulation-optimization model results

The simulation-optimization outcomes should be compared in order to establish how the model has behaved. The comparison will be performed from two points of view. First, the simulation-optimization model will be tested against a dispatching rule known as Shortest Processing time rule (SPT). Second, the comparison looks for evaluate how good it would have been to have the simulation-optimization model in the past. This can be achieved by comparing the backtesting data with the proposed model.

The dispatching rule model was developed by redesigning the simulation-optimization model. Tanks, machines and the intern process logic were respect in order to follow the Hybrid Flow Shop environment in the chemical department. Moreover, the attributes of the agents were not changed, except for the products to which a new attribute was added. This attribute is called average processing time and it is estimated by calculating the approximately time that a batch last in the machine and summing it to the time that it will take to be mix. Also, greedy algorithm was considered for ensuring a proper tank assignment. However, Local Search algorithm was eliminated. Hence, it no longer exists any diversification or intensification criteria.

The evaluation of the models utilizes real information from Fuller Pinto's company specifically of the products requested on the last five campaigns. Each campaign was tested in both models ten different times. Moreover, the Appendix-Table 8, Appendix-Table 9 and Appendix-Table 10 represents real production information such as the throughput rate, execution time and setup time of machines and tanks that is considered by both models. As a result of the simulation-optimization model evaluation, Table 9 shows that the weighted tardiness magnitude varied between campaigns due to the quantity of products that each of them had. When campaigns had a higher value of products, the weighted tardiness was greater, however, when campaigns had less orders the value of the indicator was smaller. Figure 13 illustrates the highest and lowest value of the weighted tardiness for each campaign between all ten tests.

Campaign	Quantity of products (units)	Maximum value of weighted tardiness (Ticks)	Minimum value of weighted tardiness (Ticks)
Campaign 1	86,890	28,265,000	21,458,000
Campaign 2	166,325	3,371,978,000	2,033,468,000
Campaign 3	71,618	80,981,000	46,605,000
Campaign 4	53,310	18,846,951	18,008,355
Campaign 5	182,677	5,397,862,000	2,835,587,000

Table 9. WT magnitude variation between campaigns

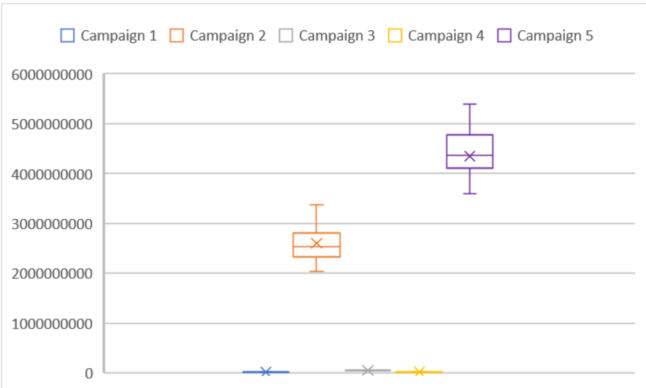


Figure 13. WT highest and lowest value for each campaign

4.4.3.2. Dispatching rule SPT results

The scheduling problem has been resolved with many methods, such as: dispatching rules, exact methods, heuristics and meta heuristics. Currently, the way of sending orders for production in the chemical area of Fuller Pinto is based on the experience of the leader of the area. In some many cases the orders that are made at first place, are the ones who can be gotten rapidly in order to make a better distribution of the rest of the time. This is the reason why the Shortest Processing time dispatching rule was selected. This dispatching rule establishes that the job with shortest processing time must be processed earlier.

A redesign of the simulation-optimization model was done by establishing that orders with less average processing time would be processed first and eliminating the Local search algorithm. As the simulation-optimization case the last five campaigns were evaluated and each campaign was tested ten times to obtain the weighted tardiness and the supply percentage. However, in this case Figure 14. Results with no strong variability illustrates that the results did not present stronger variability. One reason of the previous statement is that this model is not allowed to move between different search regions by not having any intensification and diversification criteria. Also, Table 10. WT and Orders quantity Increment demonstrates an increment in the value of the weighted tardiness as the numbers of orders evaluated increase.

Campaign	Quantity of products (units)	Maximum value of weighted tardiness (Ticks)	Minimum value of weighted tardiness (Ticks)
Campaign 1	86,890	26,365,000	21,421,000
Campaign 2	166,325	2,259,504,000	1,616,911,000
Campaign 3	71,618	72,178,000	46,796,000
Campaign 4	53,310	17,948,405	15,628,395
Campaign 5	182,677	2,995,603,000	2,734,467,000

Table 10. WT and Orders quantity Increment

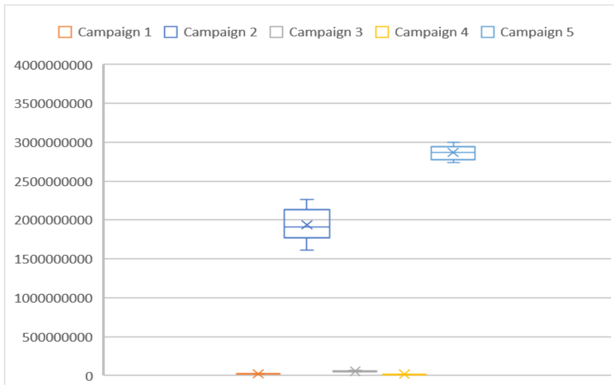


Figure 14. Results with no strong variability

4.4.3.3. Backtesting

Besides the previous, the backtesting validation method consists on a simple comparison between the real and the simulated production. First, the supply percentages of the last five campaigns of the chemical department of Fuller Pinto's company are identified. Consequently, the same campaigns are evaluated with the simulation-optimization model to similarly get each supply percentage. Finally, the real supply percentage for each campaign are compared with the "simulated" result.

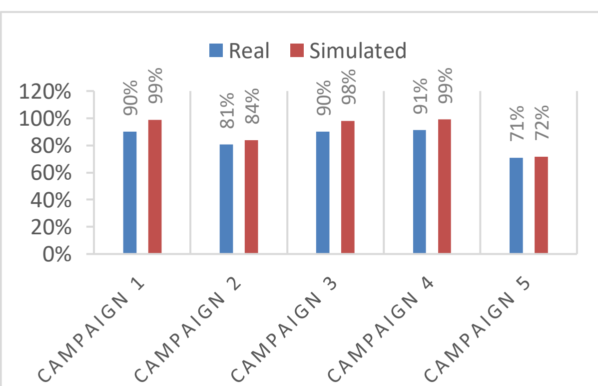


Figure 15. Real supply percentage vs simulated supply percentage

Figure 15 demonstrates how the simulation-optimization model overcome the reality by improving the supply percentage of the campaigns evaluated. In the real case, the supply percentage was always less than the goal which is reach a supply percentage higher than 98%. The results of the simulation-optimization model show a variation between itself and the reality. Consequently, the performance indicator of the campaign one was greater in nine percentage points. Also, in campaign two it was improved by a 3% and in campaign three, four and five, the supply percentage was increased by an 8%, 8% and 1% respectively.

Even though the supply percentage was improved in all cases, in some campaigns the variation between the reality and the results given by the model was not significant to reach the goal. The results are divided in two cases. The first case is campaigns which improved the supply percentage at least in a 5% and accomplished the 98% of produced orders and in the other case are the campaigns which had a variation of the indicator less than 5% and could not reached the production goal. Campaigns associated with case 2 have in common that they manage a greater number of products than the ones in case 1. A supposition that can be made is that as the quantity of products increases the supply percentage decreases.

4.4.3.4. Comparison between simulation-optimization model and dispatching rule

An experimental design was carried out to identify the relationship between the dispatching rule and the simulation-optimization model. The experimental protocol is described below.

Experiment objective

The main goal of the experimental process is to compare the effect of specific treatments in the response variable using an Analysis of Variance (ANOVA). In this specific case, the response variable is the weighted tardiness. The proposed design considers two factors and one block. The two factors considered were the *methods* and the *percentages* that set the due date back. The levels for these factors are (*simulation-optimization, SPT*) and (*100%, 85%, 75%, 65%*) respectively. It is also helpful to use the block technique to eliminate a known source of variation. In this case, the *campaign* is a known source of variation, since the orders quantity on each campaign strongly variates. The levels of the *block factor* are the last five campaigns considered.

Materials

The experimental process needs the NetLogo files of simulation-optimization model and SPT dispatching rule. In addition, the company data of the last five campaigns with orders quantities is required. The use of laptops and desktops is essential to execute the files of different campaigns with a different method. In this case, the High-Performance Computing (HPC) service of the School of Engineering of Pontificia Universidad Javeriana (ZINE) provided the services of high capacity servers. This service allowed to run the more complex simulations in terms of computational time.

Method

This method includes a few steps. First, each campaign data orders are evaluated ten times. Each time the campaign runs corresponds to one instance, until the ten runs are completed. It also exists a limit in the number of iterations that each instance must follow (100 iterations). Not every instance will complete this limit number due to the stop criterion. The release order on each iteration will be randomly set. For each iteration required, the response variables are saved. This iterative process is done for the last five campaigns with both simulation-optimization model and dispatching rule SPT. The NetLogo output data of each instance generates the weighted tardiness considering the different due dates. Finally, the values considered to build the ANOVA were the minimum weighted tardiness found on each instance data collected.

Controls

The ANOVA analysis is based on three main assumptions. The reliability of the ANOVA results depends on the fulfillment of these assumptions. The analysis assumes the independence, normality and homogeneity of variances of the residuals. For the normality assumption, the Shapiro-Wilk and Kolmogorov-Smirnov tests, the p-value result is 0,1814 and 0,5586 respectively. The conclusion for the normality assumption with a confidence of 95% is that residuals fit to a Normal Distribution. In addition, the homoscedasticity assumption can be graphically validated if the residuals of each factor are constant between levels. The associated graphics of residuals and factors that validate the homoscedasticity assumption are in the Appendix-Graphic 1, Appendix-Graphic 2 and Appendix-Graphic 3. In a similar way the independence assumption can be evaluated relating the residuals to the order in which the data was collected. The Appendix-Graphic 4 shows the residuals located randomly through all the graph. Since no pattern can be seen, the independence assumption is also satisfied. Finally, it is important to mention that all the assumptions and the ANOVA were tested using the statistical software *SPSS* and *Statgraphics.*

Data interpretation

Gathering the previous results, a Two-way ANOVA with one block can be consolidated. The result ANOVA with 5% of significance considered is presented on Table 11.

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*Fcrit*
Methods	0,591	1	0,591	2,039	**0,156**	3,9290
Percentages	64,277	3	21,426	73,882	**< 0,0001**	2,6887
Campaign	351,759	4	87,940	303,244	**< 0,0001**	2,4558
Methods*Percentages	0,533	3	0,178	0,612	**0,609**	2,6887
Error	31,320	108	0,290			
Total	448,479	119				

Table 11. ANOVA result with 5% of significance

From the ANOVA Table 11 can be established the most influential factors. The contribution of each factor is determined by calculating the p-value for each factor. Since the p-values of *percentages* and *campaigns* are less than 5%, these factors provide significative information to explain the variation of the response variable. However, the *methods* and the interaction between the *methods* and *percentage* factors does not have a significant effect on the response variable. An important measure frequently used to conclude the total variation explained by the model variables is the *adjusted* R^2. For the proposed model, the selected factors explain the 92,3% of the variation of the model. This result is a useful indicator to demonstrate that the model variation explained by the error is very little because the 92,3% of the variation is due to the selected variables.

It is important to highlight that the ANOVA analysis only stablishes that at least a pair of levels on a significant factor are different between each other, however it does not say which of the levels present this difference. For the purpose of a deeper analysis, the LSD test was performed with all possible pairs of means of each influential factor. The LSD test results are in Appendix-Table 13.

The LSD test evaluates if the difference in absolute value of SPT model mean and simulation-optimization mean is greater than the LSD statistic. If it is bigger, the null hypothesis is rejected. For the *methods* factor, the LSD test is not evaluated, since it is not a significant factor because the *p-value* in the ANOVA is bigger than 5%. As a result, it can be concluded with 95% of confidence that the null hypothesis cannot be rejected. Therefore, it stands out that the proposed model behaves at least as well as the dispatching rule. Similarly, the *percentages* and *methods* interaction did not obtain a significant influence on the response variable. The absence of interaction can be graphically seen on Appendix-Graphic 5.

Otherwise the LSD test concludes that each possible pair of *percentages* levels difference is significant. The difference between percentages clearly proofs that the tardiness mean variation is different if the due date is moved a few days before. This variation is especially high if the due date is moved from *100%* to *65%.*

Likewise, the campaigns differences are significant. As it was expected, the campaigns with lower order quantities are not significantly different between each other. For instance, the campaigns 1,3 and 4 does not reject the hypothesis of equal means. On the other hand, the campaigns 2 and 5 are strongly different between each other and between the campaigns 1,3 and 4. Once again, this result was previously expected and now confirmed, because of the considerable difference of products ordered on each campaign.

4.5. Module 5: Decision support system User Interface functionally settings

This section covers the design of the user interface for the production scheduling in Fuller Pinto's Company. The objective of the interface is to acquire the required information in order to send it to respective programs of simulation-optimization, processing the proposed approach and returning the results for the fuller pinto decision maker. For this section, a research was carried out on the requirements given by the company and the stipulated global standards that must be fulfilled. First, the requirements set in the QFD diagram were evaluated in order to create a diagram with the respective specifications. Second, the graphic interface was coded using Python as the programming language. The interface inputs are the Fuller Pinto campaign orders, including the quantity and the presentation of each product. The code structure has three main functions: Generate a CSV file, design the interface structure (with the appropriate graphics, images, entries, buttons, geometry, tittles, labels, "combo boxes" and scrollbars) and generate a Gantt chart as the output file. Third, the connection with VBA software was coded in order to send the output file with the campaign information and hide open Excel, running the VBA macro automatically. Finally, a loop was programmed in order to continuously check the folder where the JAVA CSV output file is located. Only when the file is detected, the Gantt Chart will be executed in the Python program automatically and generate the Gantt Chart with results.

The design of an application must follow some requirements in order to be successful. First, the background has to be fixed and permanent representation of a specific context of action. The interface needs an ergonomic design through the establishment of menus, action bars and easily accessible icons. There has to be an internal coherence between typography and color treatment, as well as the object of interest must be easy to identify. The GUI has to have rapid, incremental and reversible operations, with immediate effects. (Salazar-Guerrero, 2017). The interface is carried out through making use of a Mode Based Engineering System (MBSE) approach, the formalized application of modeling to support system requirements definition, design and analysis activities. Grossetti (2018). Six perspectives were followed (namely System, Needs, Traceability, Maintenance, Deployment and Operational) in order to represent the system architecture. Before sketching screens, choosing and laying out controls, cutting foam prototypes, or writing code, the development of the GUI should fully define concepts and their relationships. Software user interfaces should be designed from the users' point of view and their goals. Software systems should not distract users from their own tasks and goals, it should facilitate learning. To be perceived by users as responsive, interactive software must reduce time-consuming operations and delayed feedback for button-presses, scrollbar movement, or object manipulations.

The graphic design and layout should include, status indicators, mode indicators, prompts for input, results, error or status messages and controls. Placing important information closer to the center of the viewer's visual field, using color to highlight, boldness, density, saturation, graphic and symbols, improves its visibility and legibility. (Moretti, Marsland & Lyons, 2011) describes the experimental evaluation of an algorithmic technique that applies color harmony rules to the selection of color schemes for computer interfaces. UI was created also, having as reference this article which defines the producing color schemes that were rated most on several quality scales than those produced by random choice.

Requirements	Person #1	Person #2	Person #3
Show information easy to understand ,learn and use	✓	✓	✓
Production orders display	✓	✓	✓
Capable of including and enduring all the products of the campaigns in the data base	✓		
Show results (Production Scheduling)	✓	✓	
Introduce different presentations of the product references	✓		
Add different campaigns	✓		
Practicality to enter the data	✓	✓	
Speed and duration of the process	✓		
Easy to send or print results	✓	✓	✓
Data security	✓	✓	

Table 12. UI Requeriments

Aside from the previous research, a study was developed of the essential requirements that the company asks for the interface to be complete. Table 12 summarize results of the three people interviewed, who are part of the process of chemical production in the company. The person # 1 is the production manager who is responsible for the entire campaign to be produced on time. Actually, he needs to schedule the Chemical production in an Excel sheet with his experience. The person #2 is in charge of the personnel monitoring, controlling that they are doing well de jobs, and report the chief in case of any problem or unconformity. The person # 3 is a plant operator, who has been in the company for over 10 years and knows perfectly the production processes.

It was performed a desktop application because there no need of using internet to accomplish the work and its more functionality and easily reproducible than Webs Apps. The current study found that the best software to work on the interface is Python. The interface was made in the version 2.7, which is commonly used for developing both desktop and web applications, facilitating the visualization. In order to develop the GUI, it was used Tkinter library. It's de-facto standard GUI (Graphical User Interface) most commonly used for Python. Other libraries were used within the program as PIL (Pyhton Imagining library), Xlwing, Numpy and Datetime.

4.5.1. Fuller Pinto User Interface main screens

The objective of Fuller Pinto UI is to create an excellent user experience centered in the user's needs (Accomplish the task with relative ease, complete the task as fast as possible and enjoy the experience) divided in screens. The user who will use the interface must be able to display the productions orders, including and enduring all the products of the campaign in the data base, interpret the results in an easy and visual way, introduce different presentations of the product references and add different campaigns. The application must be enabled only for the product manager and the production monitor. They will be assigned a username and a password, to achieve the security requirement. The general structure of the pages is: In the upper part there will be the tittles of each screen and the Fuller Pinto logo. The middle right side are located the menu bottoms (Exit and results). The most important information is in the center of the screen and the items to fill or choose are close to each other to make the digitization faster. Figure 16 show Fuller Pinto design template in a story board diagram with the user interface specifications. The Appendix-Figure 1 show better the diagram.

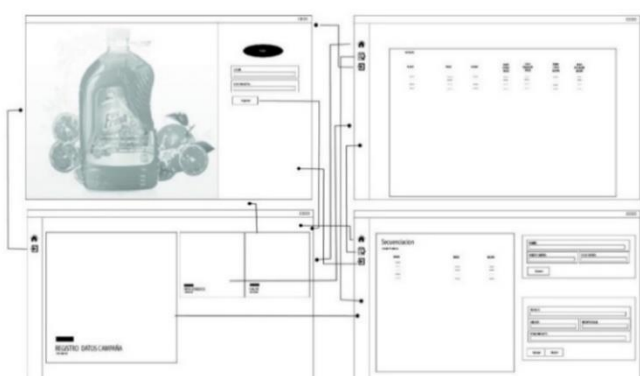


Figure 16. UI Story Board

The desktop application is divided into 3 stages: Input, internal process and output. Data input is entered by the user on the PC and have 4 main objectives developed in the internal process: Design data entries and procedures, show input data and design entries so that it's easy to fill, design data entry screens and user interface screens and use validation checks to develop input controls. Once the internal process transforms all the data into a CSV file, the Output file will show 2 different screens with the results. The output objectives are: Develop the output design that fits with the user requirements, output should be in appropriate format and make the output available on time to make a good production planning.

The interface has 5 screens; the login screen is where the user can enter with their username and password assigned. The Home screen includes the three main buttons: "Registro Datos Campaña", "Resultado" and "Salir". The Campaign Registration screen have all formats needed to fill out the orders, adding the products and its information from the campaign in order obtain the production sequence. Once the campaign runs, after a while, will appear the Gantt Chart screen. The Results screen will show more specific details of the Gantt Chart. Furthermore, since the Appendix-Figure 2 to Appendix-Figure 6 will show Fuller Pinto UI divisions. The step by step instructions for Fuller Pinto user interface are illustrated in a brochure in the Appendix-Figure 7.

4.6. Module 6: Economic Analysis

As a result of the non-compliance of the supply percentage, stockout occurs. Hence, the customers walk away or need to purchase from other sources when the product is unavailable (Gruen and Bharadwaj, n.d.). Quite often some clients can delay their purchase till the products are available again, in these cases stores back orders the products (Wang et al. 2016). In Fuller Pinto's case, the total unsatisfied demand is not given to the next campaign, a part of it is dismissed. The other part is given to the next campaign to be produced. The previous election is based on customers' requirements and it is made by the commercial department. Despite this, the economic analysis will be done taking into consideration the fact that products did not be sold due to the stockout, this will be called lost sales.

In order to estimate the quantity of lost sales, the supply percentage of each product for the last five campaigns was considered. The number of products ordered varies for each campaign. Campaigns two and five had the highest number of products, 166,325 and 182,677 respectively. However, Figure 17 illustrates that they had the minor supply percentage and the highest cost of sales in both cases: the reality and in the model. Meanwhile, campaigns one, three and four had the lowest quantity of products, 86,890, 71,618 and 53,310 respectively, the lower cost in lost sales with the highest supply percentage.

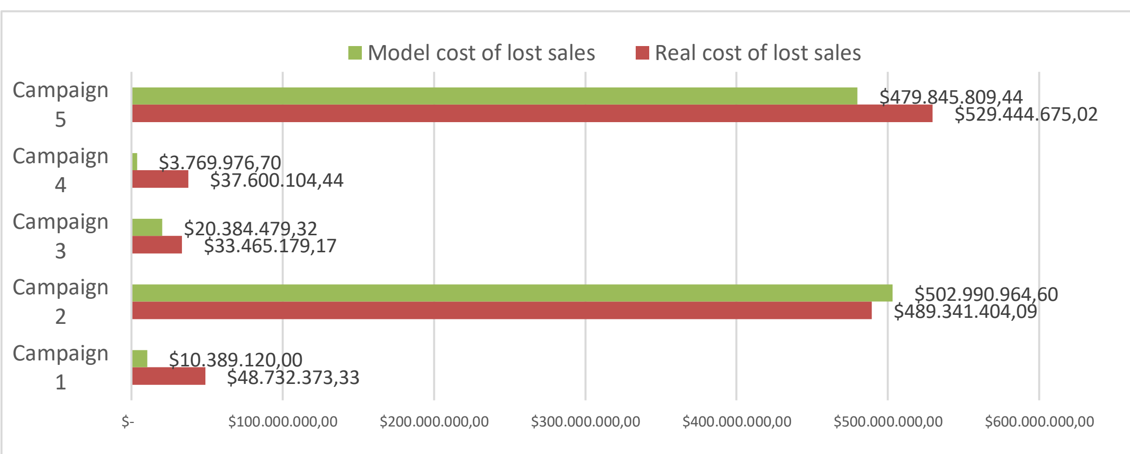


Figure 17. Real and Model Cost of Lost Sales

It is also important to mention that the simulation-optimization model does not consider minimizing the cost of lost sales. The model is continuously looking the minimization of the weighted tardiness. Even so, the reduction of the cost of lost sales compared with the reality can be a consequence of the objective of this project. The reduction of the cost of lost sales were obtained by calculating the average for each case (real case and model case) between the five campaigns. On that basis, the cost was reduced in a 11% as the cost of lost sales for the real case was $24,240,677 higher than the achieved using the decision support system.

5. Conclusions and research perspectives

This dissertation presents the jobs scheduling of the chemical area of Fuller Pinto through a simulation-optimization model. The main objective of the model is to minimize the weighted tardiness of the jobs ordered on a campaign production cycle.

A decision support system is developed to simulate and optimize the schedule of the jobs. In order to build a system according to the chemical area requirements, the QFD methodology was implemented. A sensible QFD analysis made a significant difference to meet the customer requirements. Therefore, the QFD results translated the most important customer requirements into design parameters. The conclusion of the QFD matrix analysis allowed to stablish that the decision support system design must satisfies the execution of required user tasks and fulfill the main goal of the area; obtain the optimal schedule of the ordered jobs.

Subsequently, the manufacturing environment model was developed. During this methodology phase the lot streaming process was performed first. A major conclusion of this process is the importance of a well-suited model that makes a proper batch sizing. Presenting the batch division on the mixing stage as a lot streaming problem allowed to consolidate all the tanks constraints and simultaneously look for a better resource utilization. The second methodology guarantees the previous statement, since it seeks to find a well-suited and feasible solution that allows to maximize the capacity utilization of tanks and simultaneously reducing the number of tanks used.

The second process established to represent the manufacture environment was the simulation-optimization model. From this method stands out the high capacity of modelling complex scenarios. The relevance of the simulation-optimization method is clearly supported by identifying agent's capabilities of representing the current manufacture environment decisions and behaviors. Furthermore, the simulation-optimization method is able to represent with high accuracy the complex decision-making process of a HFSP considering all the interactions and constraints of this environment. From the comparison between the simulation-optimization method and the SPT dispatching rule, can be concluded that there is no significant difference among the methods. This result supports the strength of the simulation-optimization method, since it can guarantee with a high level of confidence that the schedule will be as good as a well-known dispatching rule.

The simulation-optimization method has also shown quite benefits improving the supply percentage of the evaluated campaigns. Evaluating this indicator it can be concluded that the supply percentage policy is met in three of the five campaigns with simulation-optimization schedule, while the current method did not achieved the required percentage in any of the campaigns. In other words, the tardiness was reduced with the simulation-optimization method, increasing the supply percentage.

The historical data of the company has shown that the lack of a good resource planning and schedule results in high costs of lost sales. However, as the tardiness of the jobs is being reduced with the simulation-optimization model, the cost of lost sales is lower since it is managing to supply even more products. The absence of supplied products on each campaign is translated into lost sales.

Further research might explore the optimization problem of lot streaming. It is important to note that finding an optimal solution for the lot streaming process would affect considerably the batch behavior in the downstream stages. For improvement purposes, it is helpful to use higher computational performance. Consequently, the simulation-optimization model could run more iterations improving the search in the solutions space. Moreover, further studies could relate the developed model with the costs associated to its implementation. This research should propose a multiobjective function that involves the tardiness with the costs. Finally, a useful improvement could be done in the decision support system architecture with the goal of reduce the connections and unifying the programs required to run the system.

6. Glossary

- **Exact methods:** Algorithms that have the characteristic of using mathematical techniques, which ensure convergence to an optimal solution. (Michalewicz & Fogel, 2000).
- **Heuristic:** It is understood in the sense of an iterative algorithm that, although it generates a reasonable solution, does not converge towards the optimal or feasible solution of the problem. (Müller-Merbach, 1981).
- **Metaheuristic:** It is a high-level master procedure that guides and modifies other heuristics to find solutions beyond simple local optimality. (Glover, 1986)
- **Simulation:** Simulation is the process of designing a computerized model of a system and experimenting to understand the behavior of the system and evaluate the strategies in which the system can operate. (Shannon, 1988).
- **Optimization:** Find and identify the best possible solution, among all the potentials for a given problem, in terms of effectiveness or performance criteria. (Taylor, 1971).
- **SPT:** Delivery rule based on sequencing according to shorter processing times. (Panwalkar & Iskander, 1977).
- **Pseudocodes:** Form that allow to represent the algorithm in a block-structured language with an order in the lines. (Zobel, 2013).
- **Kruskal-Wallis test:** It is a non-parametric test that is used when the sample does not have a normal distribution or equal variance. This test evaluates the significance from each factor because is like a ONE-way anova. (Elliot & Woodward, 2007).
- **ANOVA:** Analysis of Variance. A parametric test developed from a sample with normality, constant variances and independence. This test allows evaluate the significance of the factors in the variability of a response variable. (Gutierrez, 2008).
- **Shapiro-Wilk test:** Test for check the normality assumption in samples less than 50 data**.** (Elliot & Woodward, 2007).
- **Kolmogorov-Smirnov TEST:** Test for check the normality assumption in samples greater than 50 data. (Elliot & Woodward, 2007).
- **GUI:** Graphical user interface. It allows to the users interact with the digital platforms from graphical icons. It is a human-computer interface. (The Linux Information Project, 2004).
- **Ticks Netlogo:** Unit of measurement for the Netlogo simulation, one tick is approximately equivalent to 3 seconds. (Netlogo Dictionary).

7. Appendix

The following table presents the Appendix table that contains the tables, graphics and figures attached to this document.

Table 1. Survey results of customer requirements.	Table 6. Normality assumption test for Number of Tanks performance indicator.	Table 11. Normality assumption test for weighted tardiness	Graphic 3. Homoscedasticity assumption graphical test for *campaigns* factor.	Figure 3. Home screen.
Table 2. Normality assumption test for Waste performance indicator.	Table 7. Kruskal-Wallis test for Number of Tanks performance indicator.	Table 12. LSD test for *percentages* factor	Graphic 4. Independence assumption graphical test.	Figure 4. Campaign registration screen.
Table 3. Kruskal-Wallis test for Waste performance indicator.	Table 8. Throughput rate by machine and volume.	Table 13. LSD test for *campaigns* factor	Graphic 5. Graphic of means interactions *methods*percentages*	Figure 5. Results screen.
Table 4. Normality assumption test for Computational Time performance indicator.	Table 9. Processing time by tank type.	Graphic 1. Homoscedasticity assumption graphical test for *methods* factor.	Figure 1. Diagram with user specifications. UI Story board.	Figure 6. Gant Chart screen.
Table 5. Kruskal-Wallis test for Computational Time performance indicator.	Table 10. Dependent Setup time by product type.	Graphic 2. Homoscedasticity assumption graphical test for *percentages* factor.	Figure 2. Screen Login	Figure 7. App User Manual.

References

Aurich, P., Nahhas, A., Reggelin, T., & Tolujew, J. (2016, December). Simulation-based optimization for solving a hybrid flow shop scheduling problem. In *Proceedings of the 2016 Winter Simulation Conference* (pp. 2809-2819). IEEE Press.

Barclay, Kenneth A, and W J Savage. 2004. *Object-Oriented Design with UML and Java*. Oxford: Butterworth-Heinemann. http://ezproxy.javeriana.edu.co:2048/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=195593&lang=es&site=ehost-live.

Carlier, J., & Néron, E. (2000). An exact method for solving the multi-processor flow-shop. *RAIRO-Operations Research*, *34*(1), 1-25.

Chaari, T., Chaabane, S., Loukil, T., & Trentesaux, D. (2011). A genetic algorithm for robust hybrid flow shop scheduling. *International Journal of Computer Integrated Manufacturing*, *24*(9), 821-833.

Chieh, Lin, Arthur H. M. van Roermund, and Domine M. W. Leenaerts. 2005. "Optimization Methods." In *Mixed-Signal Layout Generation Concepts*, 19–29. Boston: Kluwer Academic Publishers. https://doi.org/10.1007/0-306-48725-X_3.

El-Sherbeny, Nasser A. 2010. "Vehicle Routing with Time Windows: An Overview of Exact, Heuristic and Metaheuristic Methods." *Journal of King Saud University - Science* 22 (3): 123–31. https://doi.org/10.1016/j.jksus.2010.03.002.

Eskandarpour, Majid, Djamila Ouelhadj, and Graham Fletcher. 2019. "Chapter 11 - Decision Making Using Metaheuristic Optimization Methods in Sustainable Transportation." https://doi.org/10.1016/B978-0-12-814242-4.00011-9.

Fakhrzad, M. B., & Heydari, M. (2008). A Heuristic Algorithm for Hybrid Flow-shop Production Scheduling to Minimize The Sum of The Earliness ANDF Tardiness Costs. *Journal of the Chinese Institute of Industrial Engineers*, *25*(2), 105-115.

Ferrer, Javier, Manuel López-Ibáñez, and Enrique Alba. 2019. "Reliable Simulation-Optimization of Traffic Lights in a Real-World City." *Applied Soft Computing Journal* 78: 697–711. https://doi.org/10.1016/j.asoc.2019.03.016.

Grossetti, G., Baylard, C., Franke, T., Gafert, J., Jenkins, I., Siccinio, M., … Zohm, H. (2019). Interface and requirements analysis on the DEMO Heating and Current Drive system using Systems Engineering methodologies. Fusion Engineering and Design. https://doi-org.ezproxy.javeriana.edu.co/10.1016/j.fusengdes.2019.03.107

Gruen, Thomas W, and Sundar Bharadwaj. n.d. "Retail Out-of-Stocks: A Worldwide Examination of Extent, Causes and Consumer Responses."

International Organization for Standardization. 2011. "Systems and Software Engineering : Systems and Software Quality Requirements and Evaluation (SQuaRE) : System and Software Quality Models / International Organization for Standardization." International Organization for Standardization.

Lin, J. T., & Chen, C. M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory*, *51*, 100-114.

López-Vargas, J. C., & Marin, J. A. A. (2015). Algoritmo genético para reducir el makespan en un flow shop híbrido flexible con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia. *Entramado*, *11*(1), 250-262.

Low, C., Hsu, C. J., & Su, C. T. (2008). A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. *Computers & Operations Research*, *35*(3), 845-853.

Maynard, Harold, and Kjell B. Zandin. 2001. *Maynard´s Industrial Engineering Handbook*. Edited by Kjell B. Zandin. 5th ed. New York: McGraw-Hill.

Michael L. Pinedo. 2016. *Scheduling Theory, Algorithms, and Systems*. Fifth Edit. New York: Springer.

Moretti, G., Lyons, P., & Marsland, S. (n.d.). Computational production of colour harmony. Part 1: A prototype colour harmonization tool. Color Research and Application. https://doi-org.ezproxy.javeriana.edu.co/10.1002/col.20736

Moursli, O., & Pochet, Y. (2000). A branch-and-bound algorithm for the hybrid flowshop. *International journal of production economics*, *64*(1-3), 113-125.

Pinedo, M. L. (2016). *Scheduling: theory, algorithms, and systems*. Springer.

Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European journal of operational research*, *205*(1), 1-18.

Ruiz, Rubén, Funda Sivrikaya¸serifo˘, Sivrikaya¸ Sivrikaya¸serifo˘ Glu B, and Thijs Urlings. 2008. "Modeling Realistic Hybrid Flexible Flowshop Scheduling Problems." *Computers & Operations Research* 35: 1151–75. https://doi.org/10.1016/j.cor.2006.07.014.

Swisher, J.R., P.D. Hyden, S.H. Jacobson, and L.W. Schruben. n.d. "A Survey of Simulation Optimization Techniques and Procedures." In *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, 1:119–28. IEEE. Accessed June 13, 2019. https://doi.org/10.1109/WSC.2000.899706.

Wang, Fuqiang, Xiaoping Fang, Xiaohong Chen, and Xihua Li. 2016. "Impact of Inventory Inaccuracies on Products with Inventory-Dependent Demand." *International Journal of Production Economics* 177 (July): 118–30. https://doi.org/10.1016/j.ijpe.2016.04.019.

Yaurima-Basaldua, V. H., Tchernykh, A., Villalobos-Rodríguez, F., & Salomon-Torres, R. (2018). Hybrid Flow Shop with Unrelated Machines, Setup Time, and Work in Progress Buffers for Bi-Objective Optimization of Tortilla Manufacturing. *Algorithms*, *11*(5), 68.

Yu, C., Semeraro, Q., & Matta, A. (2018). A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research*, *100*, 211-229.

Zandieh, M., Ghomi, S. F., & Husseini, S. M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation*, *180*(1), 111-127