A MIXED INTEGER LINEAR PROGRAMMING MODEL FOR LABOR INTENSIVE MANUFACTURING CELLS CONSIDERING SKILLS AND PREFERENCES

By

Irving A. Dávila Torres

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
INDUSTRIAL ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2014

Approved by

Sonia M. Bartolomei Suárez, PhD Member, Graduate Committee	Date
Pedro Resto Batalla, PhD Member, Graduate Committee	Date
Viviana I. Cesaní Vázquez, PhD President, Graduate Committee	Date
Dámaris Santana Morant, PhD Representative of Graduate Studies	Date
Viviana I. Cesaní Vázquez, PhD Chairperson of the Department	Date

Abstract

Operators in cellular manufacturing systems are an extremely important resource since they are the ones that drive the output by putting their effort directly into the product. An operator's preference for a specific task has a significant influence in his/her performance due to its possible emotional impact. Operators with a high skill level in certain areas, but that prefer to work someplace else, can show lack of interest, consequently lowering their performance. Most cell manufacturing related literature has failed to consider non-technical factors such as operators' preferences for tasks. This work addresses this issue by proposing a three-stage assignment linear optimization model that considers simultaneously technical and non-technical characteristics of the operators and uses an objective function that measures skill-satisfaction. Moreover, the model provides cross-training recommendations that may further enhance the preference-skill combination in the system. The suitability of this model is tested using a hypothetical case study.

Resumen

En celdas de manufactura los operadores son un recurso extremadamente importante que impulsa la producción poniendo su esfuerzo directamente en el producto. Las preferencias por una tarea específica tienen influencia en el rendimiento debido a su posible impacto emocional. Operadores con un alto nivel de habilidad en cierta área, pero que prefieren trabajar otros lugares, pueden mostrar falta de interés, reduciendo así su rendimiento. La literatura en celdas de manufactura ha fallado en considerar factores no técnicos como lo son las preferencias de un operador por una tarea. Este trabajo presenta un modelo de optimización lineal de tres etapas que considera simultáneamente las características técnicas y no técnicas de los operadores usando una función objetivo que mide una satisfacción habilidosa. Igualmente, el modelo recomienda entrenamiento para mejorar la combinación de preferencia y habilidad en el sistema. La conveniencia de este modelo fue probada usando un caso de estudio hipotético.

To my parents,
Irving Davila and Teresa Torres
For all your support and trust.
Thanks for always being there for me.
I love you!

Acknowledgements

First of all I want to thank my parents for making me the conscious person of good that I am today. The way you educated and raised me, along with the values you taught me has defined my principles and personality. Thank you for all the support you have given me since kindergarten up to my college studies, even when sometimes things were difficult, all the effort and sacrifices you made were not in vain.

I want to thank Dr. Viviana I. Cesaní for working with me in this research. Thank you for all the time you took out of your busy schedule to make this happen; all your recommendations have been a key factor for the successful completion of this work. Thanks to my graduate committee, Dr. Pedro Resto Batalla, and Dr. Sonia M. Bartolomei Suárez, your recommendations were very important to improve the research work.

Thanks to Dr. Agustin Rullán, the Industrial Engineering department, and the National Science Foundation (NSF) for providing me with research assistantship during my master's studies. Your commitment to research and to helping students was an important part of my educational and professional development.

Thanks to the following professors, Dr. Cristina Pomales, Dr. William Hernández, Dr. Mauricio Cabrera, Dr. Mayra Méndez, Dr. Betzabé Rodríguez, Dr. Nazario Ramírez, Dr. Héctor Carlo, Dr. Roberto Rivera, Dr. Roberto Seijo, and Mercedes Ferrer, who have in some way contributed to my education by teaching me a class, supervising me as a teaching assistant, or providing me with recommendations for my thesis work. Thanks to the Industrial Engineering department administrative personnel, Marjorie Pratts, Kassandra Rodríguez, Mayra Colón, Israel, and Griselys Rosado, for being there all these years to help with any situation that would arise.

I also want to thank the National GEM Consortium for awarding me with the 2009 GEM MS Engineering Fellowship. Your dedication in helping underrepresented groups (African Americans, American Indians, and Hispanic Americans) has been an important part of my motivations to pursue graduate studies in order to further my knowledge.

I want to thank all the industrial engineering master students who I have had the privilege to share time with. All the fun memories and activities we shared have made this process an unforgettable one; it wouldn’t be the same without all of you. And last but not least, I want to thank Nichole Lugo for always being there for me during this process. All your support and help was invaluable to me, thank you!

Table of Contents

List of Figures

List of Tables

Chapter 1

1.1 Introduction

Nowadays, new approaches arise continuously in order to increase manufacturing productivity. Numerous factors ranging from facility arrangements, operators to machine assignments, and employees' technical characteristics have been studied in the literature in order to improve this matter. Cellular Manufacturing is an essential part of manufacturing productivity and an integral part of Lean Manufacturing Systems. Cellular Manufacturing is a workplace design model that seeks to take full advantage of the similarity between parts, through standardization and common processing. Since its focus is to process similar parts, employees in these manufacturing environments tend to have similar technical skills in their work areas, consequently bringing a greater flexibility when it comes to assigning operators to different tasks.

Many manufacturing cells are designed as small dual resource constrained (DRC) "job shops", which are systems where the number of machines exceeds the number of workers, and both worker and machine capacity constrain the output of the system (Yue et al. 2008). Even when the flexibility is an advantage in manufacturing environments, it's important to mention that functional specialization of workers may support the efficiency of cells, especially when human tasks are complex. Figure 1 shows the concept of a DRC manufacturing cell, in this cell three operators have to share the load of 10 stations in order to satisfy the required cell demand.

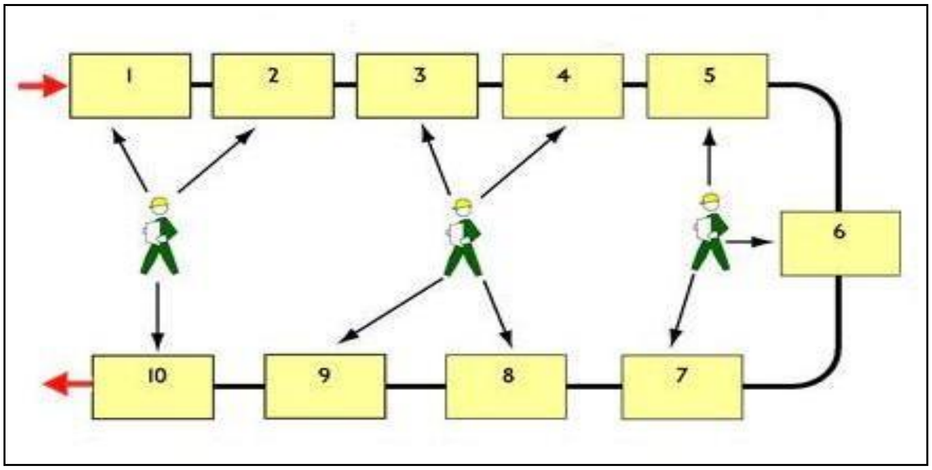

Figure 1. Cellular flow flexible layout

While extensive research has been done in order to create better designs in cellular manufacturing, an important factor that needs more attention is the operators working on it. They are the ones that drive the output by putting their effort directly into the product. Factors as simple as moods, skills, and preferences, can highly affect the output in a production line. For example, most people would think that assigning the operator with the highest skills to a station would result in a higher productivity compared to others. This is not necessarily true, since preferences may have a significant influence in an employee's performance due to their emotional impact. An operator with a high skill level in certain area of work who prefers to be assigned to other areas can show some lack of interest, thus lowering his/her performance.

The purpose of this work is to propose a method for achieving productivity in cellular manufacturing systems considering in the labor assignment process both technical and non-technical characteristics. This has been accomplished by developing a model that assigns operators in cellular manufacturing systems, while taking into consideration the skills and preferences of the operators.

1.2 Problem Definition

A great variety of models have been created in order to assign operators to stations while maximizing or minimizing different metrics in cellular manufacturing environments. Some of these metrics are costs, operators' skills, and efficiency. While most of the extensive literature directed to cell manufacturing seeks to better explain its functionality and finding ways to make cells performance efficiently, it has failed to take into account an important characteristic of an operator's performance when making assignments. This important characteristic is the operators' preference for a specific task. Because of the lack of attention paid to this factor, this document emphasizes in considering the operators' preferences when assigning them to certain tasks in a cellular manufacturing environment.

The focus of this work is to take advantage of the nature of the cellular manufacturing environment, by creating an assignment linear optimization model that satisfies product demand, while also allowing the employees to be as satisfied as possible with their assignments. The model assigns available operators to cell stations, while taking into consideration the characteristics of the cells, stations, and operators, in order to assure that the requirements of each cell and station are satisfied. Besides satisfying the requirements of the different cell stations, a preferences-skill combination metric of the operators is maximized to the greatest extent possible in order to assure that they will be working in the most comfortable/skill possible way. Furthermore, assignments to bottleneck stations are made by mainly considering the skills the operators' possess in the bottleneck station, but if there is more than one operator with the highest skill in the bottleneck station, among those operators the model will select the operator with the highest preference. Additionally, the model can provide cross-training recommendations that may further enhance the preference-skill combination in the system. Figure 2 shows the system network of the problem being addressed in this work, where s_{ijk}, p_{ijk}, and A_{ijk} represent the skill, preference, and assignment. Each operator, cell, and station are represented by the indexes i, j, and k.

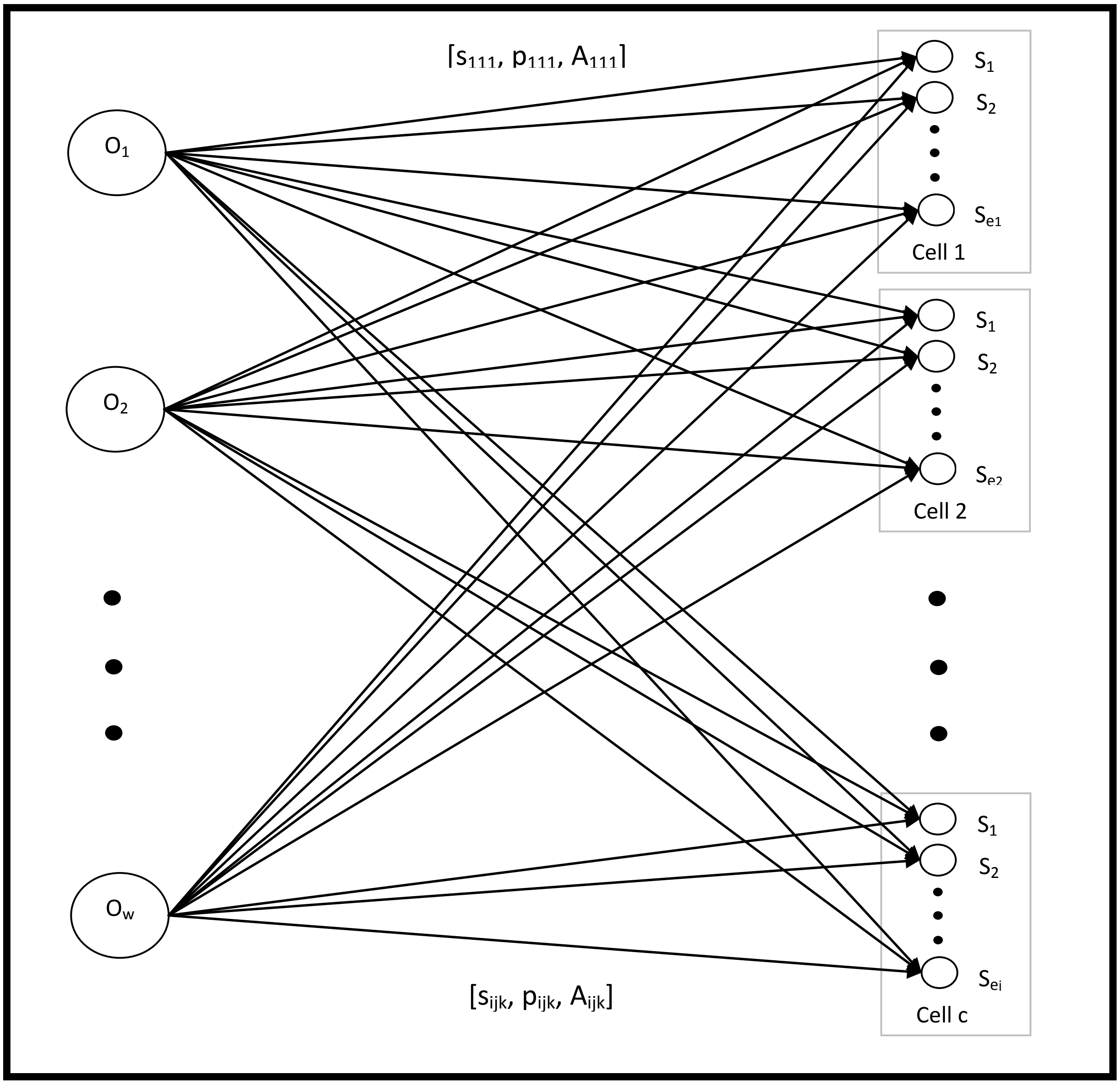

Figure 2. Intercell Model's system network representation

There is a fixed number of operators to be assigned to cell stations, and each assignment will be selected depending on the operator's preference and skill related to the assignments. The network shows the basic intercell case where operators are allowed to work at more than one cell in the system, and there is a greater range of possible assignment combinations to improve the objective function. An alternate case would be the intracell system, where operators are only permitted to work at one cell, and the possible number of assignment combinations is limited by cell. Figure 3 shows the network for the intracell case scenario policy.

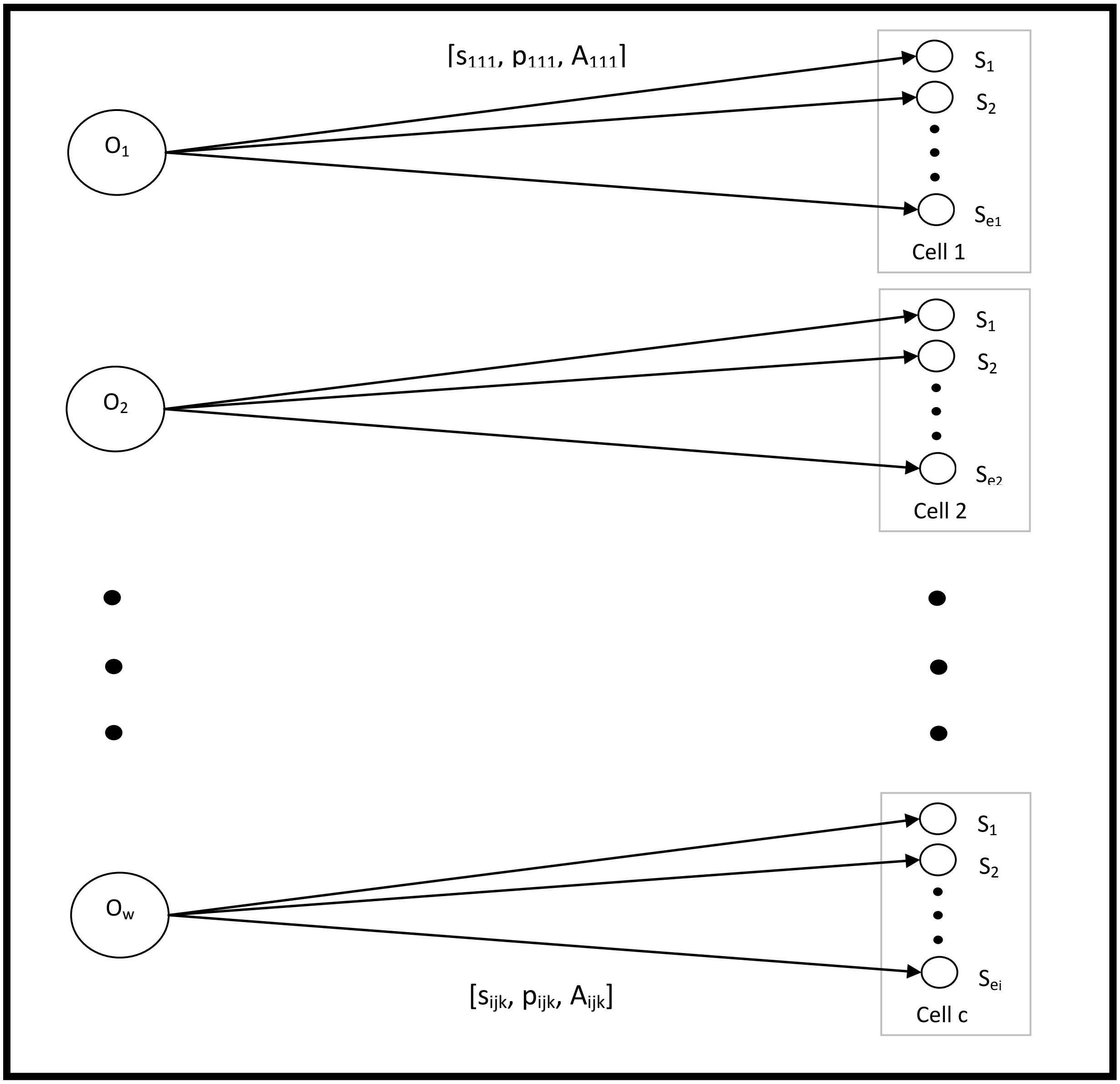

Figure 3. Intracell Model's system network representation

Essential factors that are characteristics of these manufacturing environments are taken into account in the development of the model, all of them with the purpose of creating a model whose constraints represent to the greatest extent those found in real life settings. These factors include bottleneck and non-bottleneck stations, assignments policies (intercell/intracell mobility), quantity of operators needed, operators' skills, and operators' preferences.

With regards to bottleneck stations, these are stations that limit production in comparison with the rest of the stations inside a cell. Since cells operate as independent manufacturing environments, each cell contains its bottleneck station. It is easy to see that this type of station needs a more detailed attention when considering which operator will be assigned to it. In order to address this issue, if a station is a bottleneck, the model created will consider mainly the operator's skill at the bottleneck station. If there is more than one operator with the highest skill, among them the model will consider their preferences and assign the operator with the highest preference in that station. On the other hand, if the station is a non-bottleneck station, operators are assigned by considering both skill and preferences at the same time. By using preferences and skills the model assures that each assignment contains the best possible combination of skill and preference in order to create the best skilled/satisfied system possible.

It is important to mention that the problem being addressed in this document is the assignment of operators, and it is not the scheduling or order of the different tasks that these operators will perform. For this reason, when an operator is partially assigned to several stations, it is understood that the operator will be working a certain percent of the time in a station and the rest in another station(s). The problem of scheduling, or operational sequence in which the operators will work in the stations is not addressed here, but it should be considered as a strong topic for future research.

An aspect unique to cellular manufacturing that is important to this document is that the nature of cellular manufacturing lends itself to employee cross-training. That is, employees can often be trained on multiple stations within a cell to allow for more flexibility and to account for employee absenteeism. Due to this aspect of cellular manufacturing, it is assumed that the number of operators needed in each cell is less than the amount of stations in the cell. This means that operators are required to work among several stations in a cell in order to satisfy cell demand.

Operator's skills are one of the most important features in a cell system. For the purpose of this research it is assumed that there is a company database, typically referred as the cross-training matrix that contains records of the operator's cross-training history. These records are kept by the cell supervisors and specify operator's skill based on training and experience in the different stations. These operators' skills are compared with a minimum required skill established for each station. For a successful assignment an operator must have at least the minimum skill required to perform satisfactory in that station.

Finally, the operators' preferences in the different stations are taken into consideration through interviews or questionnaires. These records are kept by the supervisor and are typically refer to as the preference matrix. This information is used with the intention of maximizing the assignment skill-preference of each operator at its station while satisfying all the constraints mentioned pertaining to bottleneck and non-bottleneck stations, assignment policies, and skills requirements in the system.

By using information about employee's skills and preferences for each station, the model assigns the most skilled employees to the most important assignments (the bottlenecks), and the remaining assignments are based on the employees' skill and preference combination related to the station. Since every assignment skill is maintained at or above the minimum skill required, it is guaranteed that production will be maintained at acceptable levels, and then enhanced with the high preferences obtained in the assignments. There are numerous reasons for a manufacturing plant wanting to maximize the job satisfaction of its employees. Obvious reasons include reducing employee turnover and absenteeism, and more importantly enhancing production, which are achieved by taking into consideration employees' preferences and skill levels as part of the assignment problem.

1.3 Objectives

Considering the research background introduced in the previous section, the objectives of this work are divided as general and specific objectives. General objectives are the ones that represent the principal goals of this work and thus, the reason for embracing this research topic. On the other hand, the specific objectives are goals with the purpose of supporting different aspects of the environment where this work is emphasized. These objectives will bring a more realistic modeling with respect to the reality of a cellular manufacturing environment. The general objectives are presented next, followed by the specific objectives.

1.3.1 General Objectives

- Develop a Linear Programming Model that addresses both skills and preferences at the moment of assigning operators to stations in a cellular manufacturing environment.

As discovered by the extensive literature research done for this work, there is a lack of work related to the inclusion of preferences within the assignment of operators. For that reason, the main goal of this work is to successfully implement the skills and preferences as decision criteria for the assignment of operators in a cellular manufacturing environment.

- Develop a method that enhances the Assignment Linear Model results, by providing cross-training recommendations for management.

The proposed model assumes that the stations at the different cells have different and predefined skill requirements. Correspondingly, operators have skills associated to the different stations in the system. If for some reason, an operator cannot comply with the minimum required skill at a station, the chances of having possibly higher objective function values would be reduced due to the lack of flexibility. In this case, our goal would be to evaluate where the operators have the greatest interest in further developing their skills, in order to provide cross-training recommendations and maximize the total system skill-satisfaction while complying with all the constraints in the model.

1.3.2 Specific Objectives

- Develop a Linear Programming Model that offers priority to stations classified as bottleneck stations at the moment of assigning operators to stations in the cellular manufacturing environment.

Bottleneck stations limit production in comparison with the rest of the stations inside a cell. Taking this matter into consideration, bottleneck stations are assigned to the operator with the highest skill on that cell station. On the contrary, non-bottleneck stations will have assignments that consider both skills and preferences.

- Develop a Linear Programming Model that offers the advantage of choosing between intracell (within the cell) and intercell (among cells) movement, giving the user the opportunity to analyze the model results for two different assignment policies.

The capability to choose between two different assignment policies provides the user with a wider perspective that result in improved and more precise management decisions with respect to operational policies.

1.4 Scope

Manufacturing companies use many different approaches to set their internal operational policies. When a specific manufacturing system is modeled, it is important to know which operational policies are in place to make sure that model results are aligned with reality. For example, if a company does not allow operators to be assigned to more than one cell, this should be considered by the model in order for the results to follow the company's policy and obtain assignments that represent the company's reality. In cellular manufacturing systems, labor flexibility is classified depending on the operator's mobility, and in this work two types of labor flexibility are considered; intercell labor flexibility and intracell labor flexibility. The use of these two policies can significantly change the results given by a model.

Intercell labor flexibility refers to the transfer of operators between cells, whereas intracell flexibility relates to operator's transfers between machines within a cell. In other words, when operators are assigned to stations in an intracell manufacturing policy, these cannot be assigned to more than one cell. Under this policy operators can work in different stations, but all these stations have to belong to the same cell. On the other hand, when a company follows an intercell policy, operators can be assigned to more than one cell in the system.

This work includes an option for the user to choose between an intracell and intercell movement policy. If the user wishes to use an intercell movement policy, it has the option to set a maximum amount of cells for operators to be assigned. This capability provides the advantage of comparing results using two different scenarios and at the same time provides a model that is adequate to a wider range of companies. Besides giving the opportunity to compare two different scenarios, this tool can also help managers on making decisions corresponding to which kind of policy would be more appropriate in order to enhance the operators' assignment skill-satisfaction. Figure 4 shows a representation of the work scope, which is mainly directed to cellular manufacturing environments. This work is directed to model the policies and assignment characteristics inside these types of environments.

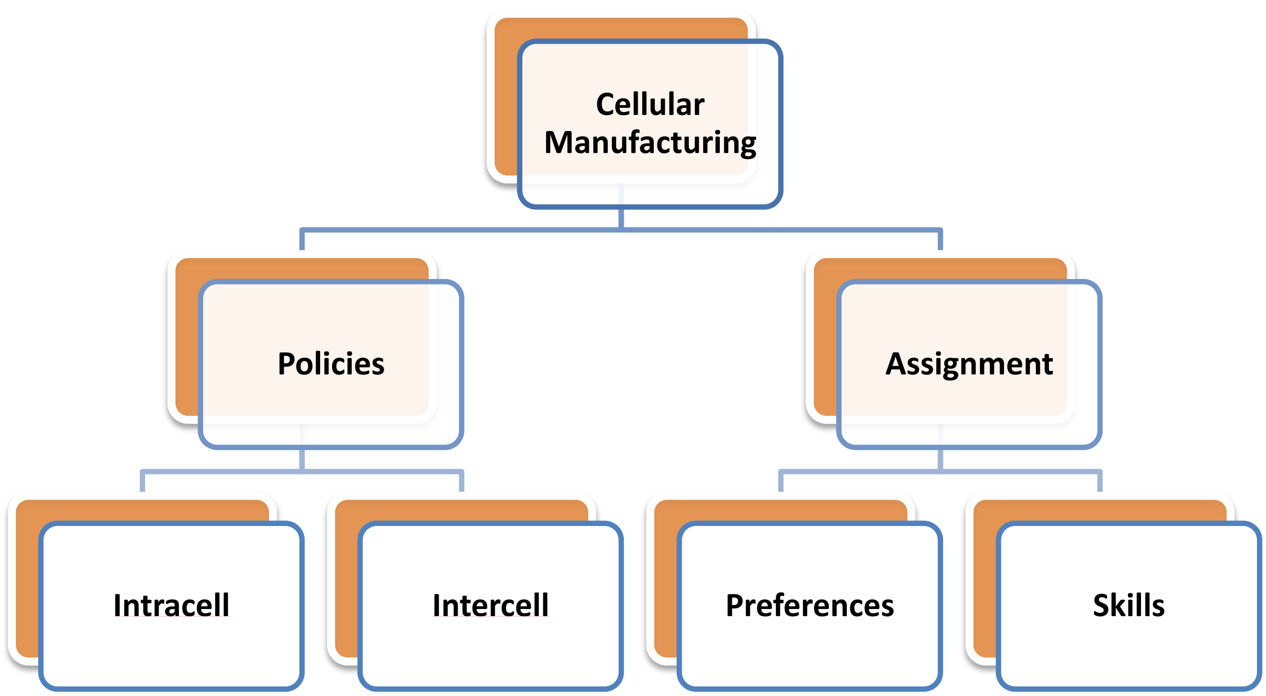

Figure 4. Research Scope

In order to add the capability to choose between an intracell and intercell policy in this work, it was noticed that an upper limit in the number of cells each operator is assigned had to be implemented into the model constraints. Keeping this in mind and with the purpose of making a user friendly application interface, a constant was added to the model and is managed through the user interface. The constant is directly passed by the interface to the model in order for the model to take it into consideration.

Cellular manufacturing systems can be classified as machine intensive and labor-intensive. In a machine intensive cell, the number of machines is the primary parameter used on determining the output, so the impact of labor on the output is limited (Egilmez & Suer, 2011). Normally, the operator's role in machine-intensive cells is limited due to the presence of automatic machines. On the other hand, in labor-intensive cells, most of the operations require light weight and small machine and equipment where the operator is continuously involved in the process, thus the number of operators and their assignment to operations has a great impact on the cell's production rate (Suer & Alhawari, 2013). Labor-intensive manufacturing cells can be found in several manufacturing systems such as food, jewelry and shoe manufacturing, medical devices, and apparel industry.

Labor-intensive manufacturing cells consist of simple machines and equipment that require continuous operator attendance and involvement. Operators are often re-assigned to different machines when a new product is released to the cell. The main reason for this re-assignment is to maximize the output rate of the cell by balancing the flow of products through several machines with varying capacities. The scope in this work aims to represent labor intensive manufacturing cells.

Since the problem to be approached has been properly introduced in this first chapter, Chapter 2 presents an in depth literature review of the cellular manufacturing environment, Chapter 3 discusses the methodology implemented, Chapter 4 examines model results, and finally, Chapter 5 summarizes conclusions and identifies future research work. Additionally, a Glossary is included at the end of the document defining important concepts used in this work.

Chapter 2

In this chapter a literature review covering the most relevant areas in the cellular manufacturing environment is presented. Some of the research areas mentioned and revised are cell formation, labor and part-machine grouping, cell size and cell loading, design and scheduling, product-sequencing, workforce cross-training, labor assignment policies, assignment of tasks to workers, and human issues in the cellular manufacturing environment.

2.1 Literature Review

In order to do a satisfactory assignment of operators there must be a good cell formation at the manufacturing environment. Concerning this issue Hoo & Moodie (1996) and Liu et al. (2009) proposed different solutions to this problem. They proposed a solution procedure for solving cell formation problems in flexible processing and routing manufacturing environments. They worked the case where a part does not have to follow a fixed path to visit some predetermined machines, and where operation or tasks can be performed in many different technical ways. In summary; given the flexible process plans of all parts and under the constraints of the machine availability, the machine capacity, the capacity requirements of operations, and the demand of the parts, they determined what parts should be produced together in the same cell, what machines and how many should be purchased (if necessary), and what machines and how many of them should be assigned to each cell, so that the total cost of operation can be minimized. An assignment of machines to cells was performed just taking into account the minimization of total cost, and not considering that when having operators assigned, their human factors can play an essential role on production performance. Similarly Liu et al. (2009) proposed a cell formation algorithm that incorporates several key production factors, such as operation sequence, production volume, batch size, alternative process routings, cell size, number of cells, and the path coefficient of material flow, leaving behind once again the importance of the operator's preferences in the process.

A hierarchical methodology for the design of manufacturing cells was proposed by Suresh & Slomp (2001), which includes labor-grouping considerations in addition to part-machine grouping. The method synthesized the capabilities of neural network methods for rapid clustering of large part-machine data sets, with multi-objective optimization capabilities of mathematical programming. It was composed of three phases; in the first part families and associated machine types were identified through neural network methods. Phase II involved a prioritization of part families identified, along with adjustments to certain load-related parameters. Finally, phase III involved interactive goal programming for regrouping machines and labor into cells. The task of regrouping workers and assigning them to the cells formed was based on several conflicting labor related goal inputs provided in terms of skills matrixes. Süer & Sanchez-Bera (1998) proposed a simultaneous solution of cell loading and cell size determination in labor intensive manufacturing cells. The study performed was a multi-period analysis where decisions were made for the next several periods. The objective was to maximize the number of products that can be completed with the available capacity in all of the periods considered with the distinctive characteristic that even though there are alternative cells where a product can be processed, once it is assigned to a cell, it is required that it is assigned to the same cell in the following periods as well. This is desirable in some industries due to setup, learning curve, communication, etc.

In the quest for developing new methods for improving productivity, Kattan (1997) presented an integrated approach to the design and scheduling of alternative hybrid multi-cell flexible manufacturing systems (MCFMSs). Kattan implemented branch and bound techniques to design group technology cells, followed by a balancing of the intercell workload of GT cells which led to a system with better utilization of the machines. Finally he proposed a heuristic method for the scheduling of a family of parts with the objective of minimizing the maximum completion time of each part. The proposed heuristic by Kattan is in some way an analogous case of the assigning of operators to machines covered in this work. While Kattan focused on assigning similar parts to machines with the purpose of reducing completion time, in this work operators are assigned to machines with the goal of maximizing the skill-preference combination of the system. Even though in this work the completion time is not taken into consideration, the approach is very similar with the difference that Kattan assigned parts to machines while this work assigns operators to stations.

Süer & Dagli (2005) developed a product-sequencing problem with the objective of minimizing the total intracell manpower transfers using a three-phase hierarchical methodology to solve the problem optimally. In the second part of the paper, a machine-level-based similarity coefficient that uses the number of machines as a similarity measure was discussed, and then used during a cell loading process to minimize makespan, and also machine and space requirements. Years later Süer et al. (2009) extended the work by Süer & Dagli (2005) by including mathematical models for cell loading based on machine-level similarity coefficients and also traveling salesman approach to minimize manpower transfers.

Another important aspect in cellular manufacturing cells is workforce cross-training. A cross-trained workforce consists of (one or more teams of) workers who have (partly) overlapping skills or tasks they are able to perform (Bokhorst & Slomp, 2007). Bokhorst & Slomp (2007) embraced an operations management viewpoint on cross-training and labor assignment with time and cost as the main strategic objective. They analyzed five important aspects to consider when developing a cross-training policy: extent of cross-training, the concept of chaining, multifunctionality, machine coverage, and collective responsibility. Their results showed that within the parallel and job shop structure, equal multifunctionality and equal machine coverage are important for achieving an optimal mean flow time, while within the serial structure, more attention should be paid to the bottleneck machines by combining unequal machine coverage and maximum collective responsibility. In all the routing structures they presented, equal multifunctionality (combined with maximum collective responsibility) seems to enable a fair distribution of workload among workers.

Yang (2007) showed a diminishing benefit in cross-training more workers and each worker in more skills, and that similarly, there is a diminishing benefit in adding more machines. In particular; when efficiency loss is high, the loss of productive time is large when cross-trained workers are sent to work on their less proficient machines. Hence, when efficiency loss is high, excessive cross-training of too many unskilled workers in too many skills can increase the incidents of incompetent workers sent to the 'wrong' machines and, in turn, increase the mean percentage of tardy jobs and work in process relative to less or no cross-training. This shows the importance of operators' skills at the moment of assigning them to workstations. In addition Inman et al. (2004) realized the importance that workers skills have and proposed a training strategy called chaining in which workers are trained to perform a second task, and the assignments of task types to workers are linked in a chain. The principle of chaining proved to be valuable in prioritizing cross-training to increase the likelihood of each task being performed by a worker formally trained on that task.

Slomp et al. (2005) demonstrated that an effective cross-training situation results if workers and machines are connected, directly or indirectly, by task assignment decisions. An integer programming (IP) model that can be used to select workers to be cross-trained for particular machines was developed. The model may help in trade-offs between training costs and the workload balance among workers in manufacturing cells. It also showed that cross-training decisions in a cellular manufacturing environment should support the forming of effective 'chains' between workers and machines through which work can be shifted, directly or indirectly, from a heavily loaded worker to a less loaded worker. On the other hand, they realized that several aspects encountered in practice such as power and personal interests have not been included in the study and deserve to be incorporated into future investigations. Brusco (2008) revisited a non-linear assignment problem for allocating cross-trained workers to maximize overall utility, measured using a quadratic function of labor shortages. He developed a branch-and-bound algorithm that efficiently provided optimal solutions for problems of practical size, and then it was used to conduct a computational investigation of cross-training policies thorough a design of experiment.

Bokhorst & Slomp (2007) also studied the literature related to the design of labor assignment rules. They concluded that literature thus far has only paid limited attention to labor assignment in systems with worker differences, and more specifically, to the who-rule. With that in mind, a series of experiments were made by means of simulation. Two experiments were conducted to study the flow time effects of applying alternative who-rules. The first experiment modeled Dual Resource Constrained (DRC) systems with homogeneous labor with respect to task proficiencies, single or multilevel flexibility, and a disparity of work center loads under three levels of average labor utilization. The second experiment modeled a DRC system with heterogeneous labor with respect to task proficiencies, single-level flexibility, and a disparity of work center loads, with 60% labor utilization. The results showed that DRC shop characteristics influence the impact of the who-rule, and that the impact of the who-rule is larger under lower levels of labor utilization than under higher levels of labor utilization. Three other simulation experiments were conducted to examine the flow time effects of the when-rule, the where-rule, and the who-rule in systems with limited labor flexibility with respect to the number of machines that workers can operate. The first experiment was focused on the where-rule and who-rule in three configurations with increasing differences in task proficiency of workers. The results showed that where-rules and who-rules that base their choice on task proficiency differences result in better flow times compared to a simple First Served (FISFS) where-rule and a random (RND) who-rule. The second experiment focused on the who-rule in three configurations with increasing differences in the number of skills workers possess. The results showed that with relatively large differences in the number of skills per worker, a who-rule that assigns the worker with the fewest number of skills results in better flow time performance than an RND who-rule. The third experiment focused on the when-rule, the where-rule, and the who-rule in a configuration with a large difference in task proficiency and a large difference in the number of skills workers possess. The results showed that a centralized when-rule performs considerably better than a decentralized when-rule.

Kher & Fry (2001) showed through design of experiment that the labor assignment policies selected have a significant effect on due date performance. Contrary to much of the literature on (DRC) systems, they showed that the choice of the where rule seems to be more important to shop performance than the choice of the when rule. Therefore, operations managers in shops manufacturing orders for both vital and non-vital customers should consider labor assignment policies and labor flexibility as important issues. McDonald et al. (2009) presented a model that assigns workers to tasks within a lean manufacturing cell while minimizing net present cost. In determining how to assign workers to tasks, the model addressed production requirements to meet customer demand, skill depth requirements for tasks, varying quality levels based on skill depth, and job rotation to retain skills for a cross-trained workforce. The model generated an assignment of workers to tasks and determined the training necessary for workers to meet skill requirements for tasks and customer demand. In selecting an appropriate labor allocation strategy, Cesaní & Steudel (2005) proposed a framework to systematically compare different labor strategies based on a given number of operators. The framework consisted of a classification scheme and empirical measures, and simultaneously considered the concepts of workload sharing, workload balancing, and the presence of bottleneck operations. The experimental results suggested that the balance in the operators' workload and the level and type of machine sharing are important concepts in determining the performance of cellular implementations.

At the Naval Surface Warfare Center (NWSF) DePuy et al. (2009) discussed three heuristic approaches in order to assign tasks to workers based on skills requirements/competency profiles, and to generate a low cost training schedule to resolve current skills gaps. Although minimizing training costs is a very important objective for most companies out there, they stated that there are other factors that NSWC would like to consider as well. Such factors are taking into consideration worker preferences and manager preferences during the assignment process, thus showing us how the consideration of human issues in cellular manufacturing workers to task assignments is crucial. In order to give us a clearer view of the different types of assignment problems, Pastor & Corominas (2007) proposed a basic classification of the job assignment problem and its variations, and discussed how they have been solved in the specialized literature. Having set out the classification, the variations were divided into three types, depending on whether they consider the evaluation function of the solutions, the constraints to be taken into account in the assignment, or other practical aspects that define the problem like deterministic or random data, the required capacity for different types of task, duration of the task in the period, and duration of the period.

Hyer & Brown (2003) reported a study that observed that engineers do not always appreciate being tied directly to the shop floor in manufacturing environments. They saw how a cell manufacturing company provided an on-site tuition reimbursed MBA program, an in-house workout facility, onsite travel agency, and a 4.5 day work week benefits, all designed with the intention of engineers overcoming the resistance to the company's policy of locating them on the shop floor adjacent to the cells they support. Through this study is seen that there is always a willingness in employees that determines their performance at work regardless of the incentives given to them. Relating to performance Fitzpatrick & Askin (2005) presented a mathematical formulation of the team construction problem using a set of labor skill pools, and showed that team performance depends on individual behaviors and interpersonal interactions, as well as technical competence.

In an attempt to explore in more detail the human issues in the cellular manufacturing environment, Bidanda et al. (2005) presented an overview and evaluation of the diverse range of human issues involved in cellular manufacturing based on an extensive literature review. In addition to an extensive literature review in human issues, they made a survey to determine the importance of eight different human issues in cellular manufacturing and gave it to a sample of academics, managers, and workers involved in cellular design, leading to the conclusion that consideration of technical issues alone cannot guarantee that an organization can develop and implement an optimal cell design. The same reasoning was made by Nembhard & Norman (2007) stating that human factors such as learning ability, motivational issues, and worker attitude, should be considered in order to make studies more applicable.

Through the literature review of this document it is seen the lack of research that exists with regards to considering preferences when assigning workers to tasks. On the other hand, there has been plenty of emphasis in the skills aspect of assignments, and a number of different aspects that vary from company to company. As stated by Bokhorst & Slomp (2007) there are several social arguments for limiting labor flexibility in manufacturing cells. On the other hand it has been proved that high levels of labor flexibility may impair social identity because the different jobs in a team/cell will be more similar, what may cause motivational deficits (Fazakerley, 1976). In addition, with respect to their abilities, people may prefer diversity within the team/cell because being a specialist enhances feelings of being unique and indispensable, making the contribution to group performance visible (Clark, 1993). Moreover, studies pertaining to diversity reveal that creativity and motivation are greater in teams whose members have different, but somewhat overlapping, skills (e.g., Jackson, 1996). In these arguments we should focus on certain combination of keywords that have been continuously ignored by researchers: social, identity, motivational, deficits, abilities, diversity, specialist, feelings, unique, creativity, and the most important of them, performance. These are important words that distinguish persons from one another, and need to be considered in the job assignment problem because they are a relevant connection to the workers performance. It is like a cluster of words that come together to tell us that social identity, creativity and motivation, combined with the diversity of abilities determines feelings of uniqueness and performance of a person, or on the other hand may cause a motivational deficit. As the literature review showed, worker's preferences are directly related to workers' motivation, thus consequently impacting their performance. With all this in mind, a model for assigning workers to tasks was developed considering the operators' skills and preferences in order create an assignment that considers the human non technical characteristics.

Chapter 3

This chapter introduces the details of the model and the interface developed. A description of the model components and assumptions along with the model functionality is explained.

3.1 Model Approach

The approach for the model execution is composed of four stages. The first stage performs the bottleneck assignments. The bottleneck assignments mainly consider the skills the operators have in the bottleneck stations. Once these bottleneck assignments have been made, the second stage is executed. The second stage consists of maximizing the minimum skill-satisfaction between all the operators in the system, while leaving the bottleneck assignments fixed. Next, the third stage to be executed incorporates the minimum skill-satisfaction between all the operators obtained in the second stage as a minimum constraint, in order to maximize the total system skill-satisfaction of the system. Lastly, a fourth stage can be optionally executed in order to obtain cross-training recommendations, where the skills of the operators can be increased in order to elevate the total system skill-satisfaction. Figure 5 illustrates a flowchart of the logic that the programming in Lingo 11 executes. The flowchart shows the order in which the models are executed as well as the different decisions that the programming considers for deciding which models are executed or not. Finally, this flowchart also shows the different variables that are set to fixed values, in order to pass them from one model to another. The S, and T are the model skill and cell training variables, which change due to cross-training recommendations.

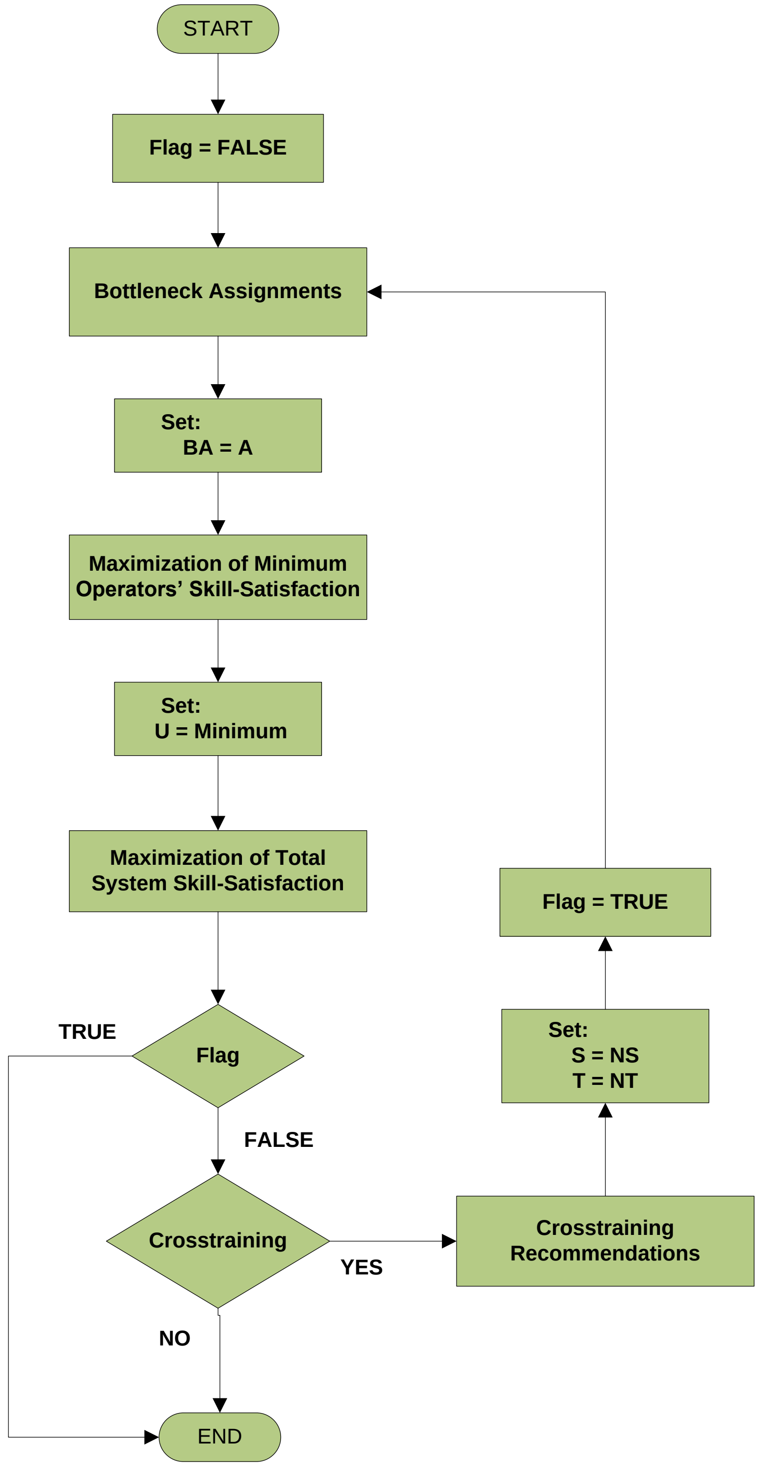

Figure 5. Model's Lingo logic flowchart

3.2 Model Assumptions

This section introduces the assumptions that the model follows in order to provide accurate results in the area implemented. There are two main assumptions that need to be followed, and are explained next:

1. Model parameters are known a priori based on company database – The performance of the model depends primarily on the parameters entered into the model. If the model parameters are not accurate the results will not provide realistic assignment recommendations. The following are the most important parameters that need to be known a priori.

    - Skill Levels – These are obtained from the cross-training matrix of the company.

    - Preferences – These are obtained through focus groups & questionnaires made by the company.

    - Demand Requirements – These are to be acquired through the product mix & demand specifications of the company (MRP System).

2. Cells and stations are located at near distances from each other – Since the model created offers the option to assign operators to more than one cell, and more than one station, we need to assure that the distance between cell and stations is close enough for an operator to travel from one to another in a considerably small amount of time.

3.3 Model Limitations

This section introduces the limitations that the model possesses. These limitations arise from the way the problem was formulated. There are two main limitations and are explained next:

1. Model applies for a system where cells perform only manual work (No Machines) – This model is designed for systems where all the operations are performed manually by the operators. Systems with automatic or semiautomatic operations are not compatible with this model because these have loading and unloading times associated with the machines. These times are not considered in the model.

2. Model is designed to be applied to DRC systems – This model was created with the purpose of addressing systems where the number of operators is less than the number of machines, and these have to share operations in order to fulfill the demand. The model results provide with how much time and where each operator is going to be assigned. If the amount of operator is the same as the number of machines, a classic assignment problem would be the recommended tool to be used.

3.4 Model Structure

This section explains the structure of the model implemented. The structure is presented in different parts, starting with the definition of sets, followed by the decision variables, relevant parameters, objective functions, and the model constraints.

3.4.1 Sets

Sets are the different groups or areas that are represented in the model. These are seen in the model as indexes associated to the different variables or attributes. There are three basic and straightforward sets that are used with the purpose of covering all those areas relevant with the assignment of operators. In the model, the indexes i, j, and k refer to operator, cell, and station respectively, as shown below. Moreover, every decision variable and attribute used in the model can be represented using matrixes that are defined with the indexes i, j, and k.

- Operator (i) = 1, 2… w
- Cell (j) = 1, 2… c
- Station (k) = 1, 2… e_j

Where $e_j = [e_1 \ \ e_2 \ \ldots \ e_c]$, represents a vector that provides the number of stations that each cell contains.

The variables **w** and **c** represent the maximum value of these sets, and should be selected prior to running the model, and depending on the scenario to be modeled. These variables represent the total number of operators and cells in the problem to be modeled. Since in the CMS environment the number of stations varies depending on a specific cell, the variable $\mathbf{e_j}$ represents the number of stations in each cell. Thus, in our model the index k maximum value will vary depending on the corresponding cell j.

3.4.2 Decision Variables

Decision variables are those that are to be adjusted by the model in order to best meet the objective function, while at the same time satisfying the model constraints. The main decision variables in this model are the assignment of each operator to the different cell stations, bottleneck assignments, minimum skill-satisfaction for all operators, and the cross-training level incurred. Other decision variables are introduced in the model with the purpose of creating constraints that help in simulating the different scenarios seen in the model. This type of decision variables are addressed as Auxiliary Decision Variables. Next, the decision variables utilized are introduced and explained in detail.

3.4.2.1 Main Decision Variables

Assignment

The assignment of operators is the most important decision variable in this model since, depending on this arrangement, is that the objective function goal will be reached. This variable represents the fraction of the shift that the operator will be working at a certain cell station. The assignment of operators is a variable that depends on both stations and cells since the model takes into consideration the assignment of operators to different cells and stations, meaning that an operator can be assigned to more than one station. The assignment variable ranges from zero to one (0-1), and is introduced below with its corresponding indexes and description.

$$A_{ijk} = Assignment\ of\ operator\ i, at\ the\ station\ k\ inside\ cell\ j.$$

Bottleneck Assignment

The bottleneck assignment of operators is a decision variable used for the bottleneck stations. This bottleneck assignment variable ranges from zero to one (0-1), and is used in the bottleneck assignment model, and then through a linear relationship it is passed to the final assignment variable A_{ijk}.

$$BA_{ijk} = Bottleneck\ Assignment\ of\ operator\ i, at\ the\ station\ k\ inside\ cell\ j.$$

Operators' Minimum Skill-Satisfaction

One of the model stages to be analyzed in this work looks to maximize the operators' minimum skill-satisfaction. In order to implement this objective function as part of a linear model, the decision variable U which represents the minimum operators' skill-satisfaction was introduced.

$$U = Minimum\ operators'skill - satisfaction.$$

Skill Training

The skill training decision variable is used for the cross-training model. This decision variable is an integer variable that ranges between zero and nine (0-9), and that states the skill level increase to be awarded to an operator in a certain cell station. A skill training of zero means that there was no skill training given, and a skill training of nine means that the skill was increased to the maximum possible.

$$ST_{ijk} = Skill\ Training\ for\ operator\ i, at\ the\ station\ k\ inside\ cell\ j.$$

Cell Training

The cell training decision variable is used for the cross-training model. This decision variable is a binary variable that states if an operator receives training to work in cell j.

$$CT_{ij} = \begin{cases} 1\ if\ operator\ i\ is\ trained\ to\ work\ \ at\ cell\ j \\ 0\ otherwise \end{cases}$$

3.4.2.2 Auxiliary Decision Variables

Cell Assignment

The operators' cell assignment is a binary decision variable created in order to comply with the cell movement constraints, which are shown subsequently. These constraints allow for an intracell or intercell movement policy to be used in the model. The variable, along with its description, is shown next.

$$Y_{ij} = \begin{cases} 1\ if\ operator\ i\ is\ allowed\ to\ be\ assigned\ at\ cell\ j \\ 0\ otherwise \end{cases}$$

Assignment Binary Flag

The assignment binary flag is a decision variable that tells if an operator was assigned to a specific cell station. It is used with the purpose of making possible some constraints. This decision variable is a binary variable specified by operator, cell, and station, as is shown below with its description.

$$I_{ijk} = \begin{cases} 1\ if\ there\ is\ an\ assignment\ of operator\ i\ to\ station\ k\ inside\ cell\ j \\ 0\ otherwise \end{cases}$$

New Assignment

The New Assignment decision variable is introduced in order to create a relationship between the bottleneck assignment and the final assignment decision variable. This variable is determined once the bottleneck assignments have been done. The new assignment variable ranges from zero to one (0-1), and is introduced below with its corresponding indexes and description.

$$NA_{ijk} = New\ Assignment\ of\ operator\ i, at\ the\ station\ k\ inside\ cell\ j.$$

Skill Training Flag

The skill training flag is a binary decision variable that tells whether an operator received or not skill training. It is used with the purpose of making some constraints possible in the cross-training model. This decision variable is a binary variable specified by operator, cell, and station, and is shown below with its description.

$$Q_{ijk} = \begin{cases} 1\ if\ there\ is\ skill\ training\ for\ operator\ i\ in\ station\ k\ inside\ cell\ j \\ 0\ otherwise \end{cases}$$

New Skill Training

The new skill training decision variable was created in order to make a relationship between the old skill and the new resulting skill after cross-training. The new skill training variable is an integer variable that ranges from zero to ten (0-10), and is introduced below with its corresponding indexes and description.

$$NS_{ijk} = New\ Skill\ of\ operator\ i, at\ the\ station\ k\ inside\ cell\ j.$$

New Cell Training

The new cell training decision variable was created in order to make a relationship between the old cell training and the new resulting cell training after cross-training. The new cell training variable is a binary variable, and is introduced below with its corresponding indexes and description.

$$NT_{ij} = New\ Cell\ Training\ of\ operator\ i, at\ the\ cell\ j.$$

3.4.3 Parameters

Parameters are the constants used in the constraints and the objective functions. Most of the parameters used in this model are characteristics that describe the different sets in the model. Among these characteristics we have the operators' preferences and skills at the different cell stations, requirement of operators needed at the different cell stations, minimum required skill for an operator to be assigned at a cell station, etc. The parameters used in this model are now explained in detail.

"Big M Method" M

The "Big M Method" M, is a constant big enough used to rule out constraints when specific scenarios occur. The way it works is by forcing constraints to be satisfied when some variable will not make the constraint possible. This constant is represented by the letter ***M***.

$$M = \text{Constant big enough}$$

Bottleneck weight factor

The bottleneck weight factor is constant small enough created with the purpose of among the operators with the highest skill, assigning the operator with the highest preference to the bottleneck station. This factor is used in the Bottleneck stations stage of the model. This constant is represented by the letter ***w***.

$$w = \text{Constant small enough}$$

Maximum number of cells by operator

The maximum number of cells by operator is a constant that should be defined prior to running the model. This constant is used in the Cell Movement Constraint, which specifies the maximum number of cells at which an operator can be assigned. This constant is represented by the letter ***v***.

$$v = \begin{cases} 1 \ for\ Intracell\ Movement\ policy \\ > 1 \ for\ Intercell\ Movement\ Policy \end{cases}$$

Maximum number of stations by operator

The maximum number of stations by operator is a constant that should be defined prior to running the model. This constant is used in the Stations Movement Constraint, which specifies the maximum number of stations at which an operator can be assigned to. This constant is represented by the letter $\boldsymbol{h}$.

$$h = \text{maximum number of stations at which an operator can be assigned}$$

Operator Cell Training

In a cellular manufacturing environment, stations inside a cell are dedicated to similar parts or products. This means that different cells may be working different kinds of products, and consequently using different machinery or techniques. Since the skill that an operator has will vary on the products or techniques utilized at the cells, skills change among different cells. To cover this aspect of cellular manufacturing, the following parameter that represents whether an operator is trained or not to work in a cell is introduced.

$$t_{ij} = \begin{cases} 1\ if\ operator\ i\ is\ trained\ to\ work\ in\ cell\ j \\ 0\ otherwise \end{cases}$$

Operator Skill

As stated before, the skill of an operator is one of the most important characteristics at the time of assigning operators to stations. The skill is a parameter that has to be defined by every company in its way. The fact that each company may define the skill of an operator in a different way makes the skills parameter difficult to be defined. For example, for one company the skill of an operator could be defined by the amount of time the operator has been working the process, or the knowledge the operator has in the process. On the other hand, other companies could define it as the productivity numbers related to the operator in the process. The point is that every company must define what the term skill represents in their particular case, and then convert this metric to a zero through ten scale, where ten is the best skill possible. The skills parameter ranges from zero to ten (0-10), where a value of zero in a cell station means that the operator is not trained to work in the cell where the station resides. By taking this matter into consideration and the fact that operators have different skill levels depending on the cell station they are assigned, a parameter that specifies the skill of an operator at each cell station was created.

$$s_{ijk} = \ Skill\ of\ operator\ i\ at\ the\ station\ k\ inside\ cell\ j.$$

Operator Preferences

Operators' preferences are of great importance in this work since our goal is to introduce the preferences in the assignment of operators to stations. Operator preferences from the different processes in the system could be achieved by distributing surveys that target the operator preferences in the system, and finally converting these surveys to a one through ten scale, where ten is the highest preference possible. With this in mind, a parameter that that ranges from one to ten (1-10), that specifies the preference of an operator by cell and station was created, and is introduced next. For more information on how to develop a survey towards gathering operator preferences in cellular manufacturing systems please refer to Robles-Leon (2013)

$$p_{ijk} = \ Preference\ that\ operator\ i\ has\ for\ working\ at\ the\ station\ k\ inside\ cell\ j.$$

Cell Station Time Requirement

In cellular manufacturing there is a quantity of time needed in order to comply with the workload at a certain cell. Operators are assigned certain amount of time among the different cell stations. This time is called the time requirement, and it depends on the standard time of each cell station, and the demand units in each cell. This time can also be called the fraction of the shift time required by operators to work at the different cell stations. This parameter introduced is calculated a priori to the model execution through known data of each cell.

$$d_{jk} = \ Time\ requirement\ in\ cell\ station\ j\ k.$$

Required Skill

In order to assign operators to stations considering the skills they have, a minimum skill required per station was established. This parameter has to be established by the company personnel through their knowledge about the difficulty and complexity of the different processes in the system. For this model, it will be assumed that these minimum required skills are known through specifications provided by the company management. This parameter is set by cell and station as introduced next.

$$r_{jk} \ = \ Minimum\ Skill\ required\ to\ work\ at\ the\ station\ k\ inside\ cell\ j.$$

Minimum Assignment

Since we are dealing with computer software, we have to state a minimum assignment permitted per cell. If that minimum assignment is not stated, the model could offer a solution in which there is an assignment of .0000001 to an operator while meeting every constraint in the model. This is a result that does not make any sense since an operator cannot be .0000001 percent of a shift working at a station. For practical purpose and because different cells are able to work different families of products, and consequently have different performance specifications, it was decided to declare a minimum assignment by cell.

$$l_j \ = \ Minimum\ operator\ assignment\ inside\ cell\ j.$$

Bottleneck Stations

It was mentioned earlier that a factor to be considered in this model is whether or not a station is a bottleneck station in order to make the appropriate dedicated assignments. This information is assumed to be known in advance through a workload analysis performed on the cells (capacity analysis).

$$b_{jk} \ = \ \begin{cases} 1\ if\ station\ k\ at\ cell\ j\ is\ a\ bottleneck\ station \\ 0\ otherwise \end{cases}$$

Cross-training Budget

In order to provide cross-training recommendations through the cross-training model, a budget in terms of dollars had to be introduced. This budget represents the amount of money that the management has available to cross-train the personnel. This budget is associated to the training costs to be later introduced.

$$B = \ Available\ budget\ \ for\ cross - training.$$

Cell Training Cost

In order to develop a cross-training model and to establish priorities, a cost for increasing the knowledge in the different cell areas is introduced. One of the possible cross-training alternatives is to train operators in new cells where they did not have previous training. This cost has to be defined by the company and represents the money required to formerly train and familiarize an operator in a new cell it didn't have the knowledge and ability to work before. For that purpose, the cell training cost parameter was introduced to represent the cost associated with this activity.

$$c_j = cell\ training\ cost\ for\ cell\ j.$$

Skill Training Cost

Associated to the cross-training model, an additional cost was introduced for the increase in the skills the operators already have in the different cell stations. This cost has to be defined by the company and can be represented as the losses in productivity obtained by exposing an operator to further higher their skill levels. Another way to represent this cost is as the prices associated to additionally train and capacitate an operator in a specific process in the system. For that purpose, the skill training cost parameter was introduced and specified by cell and station.

$$sc_{jk} = skill\ training\ cost\ for\ station\ k\ inside\ cell\ j.$$

3.4.4 Objective Function

Using the information associated with each assignment skill and preference, we can obtain the "skill-satisfaction" achieved by each operator due to its assignment, as well as the total skill-satisfaction of the system. This skill-satisfaction is chosen as the main performance metric to be analyzed because it maximizes the skill and preference combination, and at the same time represents a way of measuring a skilled feeling of fulfillment for the operators in the system. Equation 1 represents the skill satisfaction of an operator i.

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} p_{ijk}s_{ijk}A_{ijk} \ , \forall \ i \tag{1}$$

In order to achieve the model's goal, four different objective functions were created. The purpose of using four different objective functions is to be able to integrate different scenarios into the model. The first objective function is used for the bottleneck stations assignments by mainly considering the operators' skills. In case there are two operators with the same highest skill the model selects the operator with the highest preference, this because of a small weight multiplied by the preference in the objective function. Then, two more objective functions are used where the result of the second objective function is used as an input on the third one. The second objective function will maximize the minimum skill-satisfaction among all the operators, and then this minimum operators' satisfaction will be used as a minimum constraint with the goal of maximizing the total system skill-satisfaction in the third objective function. Finally, the fourth objective function tries to maximize the total skill-satisfaction by providing cross-training recommendations in the system. Figure 6 provides a flowchart with the order in which the different objective functions are executed.

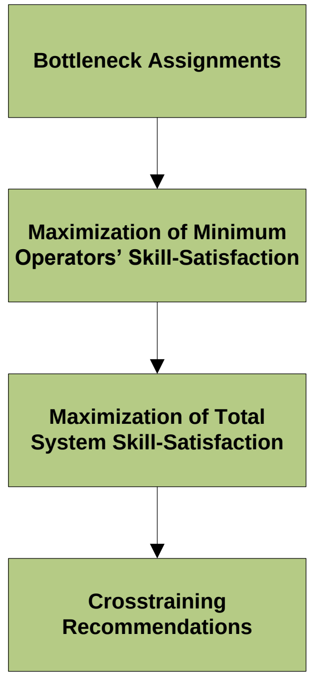

Figure 6. Flow of objectives functions in the model

3.4.4.1 Maximization of the Bottleneck Total System Skill

This first objective function is created with the purpose of assigning operators to the bottleneck stations by mainly considering the skill of the operator, and in case there is more than one operator with the highest skill, the model selects the operator with the highest preference. In order to accomplish this, the objective function is the skill plus a small weight of the preference, all multiplied by the assignment variable. The weight utilized should be really small so that in no way the preference surpasses the importance of the skill. The weight will only make a difference when there is more than one operator with the same highest skill. This objective function is maximized by choosing assignments that increase to the greatest extent possible the total sum of the weighted skills, what is shown by equation 2.

$$Max \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} BA_{ijk}(s_{ijk} + w \cdot p_{ijk}) \tag{2}$$

3.4.4.2 Maximization of the Minimum Operators' Skill-Satisfaction

The second objective function is deals with the minimum skill-satisfaction across all the operators, and this model seeks to maximize it. This stage seeks to maximize the minimum of a set of values, what makes our model a Non-Linear model. Equation 3 shows the representation of what we want to do with this objective function.

$$Max \left\{ Min \sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk}, \ \ \forall\, i \right\} \tag{3}$$

To create this objective function in terms of a linear set of equations, it was separated into two steps. It can be appreciated in equation 3 that we want to maximize the minimum of a constraint. For this reason, the first step was to create a new constraint that introduces the minimum operators' skill-satisfaction, which is shown in equation 4.

.

$$\sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk} \geq U, \ \ \forall\, i \tag{4}$$

As seen in equation 4, it is specified that the operators' skill-satisfaction has to be greater or equal than U for each operator. U represents the minimum operators' skill-satisfaction, and is introduced as a decision variable. Since the goal of this stage is to maximize the minimum operators' skill-satisfaction, we represent our goal with this objective function simply as follows in equation 5.

$$\boldsymbol{Max}\ (\boldsymbol{U}) = \ Max\ \left\{Min \sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk}\,,\ \ \forall\, i\right\} \tag{5}$$

3.4.4.3 Maximization of the Total System Skill-Satisfaction

This third stage seeks to maximize the total system skill-satisfaction. The calculation of the total system skill-satisfaction is done by summing the product of the assignment decision variable A_{ijk}, and the corresponding assignment preferences and skills. In other words, since our decision variable A_{ijk} can acquire values between zero and one, a weighted sum of skill and preferences is being calculated. The goal with this objective function is to maximize it, by choosing assignments that increase to the greatest extent possible the total sum of the system skill-satisfaction shown in equation 6.

$$Max \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk} \tag{6}$$

3.4.4.4 Cross-training Maximization of the Total System Skill-Preference

This last stage of the model tries to maximize the total system skill-preference combination. The calculation is pretty similar to the third objective function with the difference that the assignment decision variable is not included in the skill-preference product, which is the reason why it is not considered a satisfaction measure. This total system skill-preference is obtained by summing the multiplication of the preferences and the resulting new skill after cross-training as shown in equation 7. This stage will maximize the new skill on the operators with more interest on developing new skills, which later will bring a higher skill satisfaction on the system.

$$Max \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk} \, NS_{ijk} \tag{7}$$

3.4.5 Constraints

The constraints of the model are equations that force the decision variables to remain in some specific range of values with the intention of obtaining results that go according to our model's goal. The constraints utilized in this model will be introduced and explained in detail with the intention of providing the reader on understanding the logic of how each of these constraints work.

Cell Station Time Requirement

This model assumes that the time requirement in each cell station is known in advance through rough-cut calculations of demand and standard times. For this reason, it is compulsory to assign the operators the right amount of time at each cell station, as specified by the cell station time requirement.

The total quantity of time assignment to a cell station must be equal to the time requirement of the cell station.

$$\sum_{i=1}^{w} A_{ijk} = d_{jk} \text{ }, \forall\, j,k \tag{8}$$

Assignment Binary Flag Control

The binary assignment flag $I_{ijk}$was created with the purpose of identifying when there is an assignment made in a cell station for any operator. In order for this decision variable to acquire the necessary correct values the following constraints were introduced in the model. Equation 9 declares I_{ijk} as a binary variable. Equation 10 states that I_{ijk} has to be greater than or equal to A_{ijk}, so if there is an assignment, I_{ijk} will be forced to be one, while on the contrary if there is no assignment I_{ijk} can be either one or zero. The constraint Operator Minimum Assignment discussed next, forces I_{ijk} to be zero when there is no assignment.

The binary flag I_{ijk} will become one, only when A_{ijk} is greater than zero.

$$I_{ijk} \in \{1,0\} \text{ }, \forall\, i,j,k \qquad \textit{(9)}$$

$$I_{ijk} \geq A_{ijk}, \forall i,j,k \quad (10)$$

Operator Minimum Assignment

This constraint is created with the purpose of avoiding the model to provide nonrealistic results. For example, an assignment of .00001 would not make sense as a result. Also, there could be certain assignments that would not be appropriate due to these being too low in order to provide an effective and productive cell. For this reason, the parameter l_j was created to specify a minimum effective and productive assignment for each cell. This minimum assignment is assumed to be known in advance through company management. The way this constraint works is by comparing the Assignment decision variable, to the product of the minimum assignment parameter and the assignment binary flag I_{ijk}. It can be seen that the assignment variable can be greater than or equal to the minimum possible assignment only when I_{ijk} equals 1, which means that there was an assignment made in the cell station *jk* for operator *i*. On the contrary, if no operator is assigned to the cell station, the constraint states that the assignment has to be greater than or equal to zero. This constraint also forces I_{ijk} to be zero when there is no assignment.

An operator's assignment in a cell must be greater than or equal than the minimum assignment possible at the cell.

$$A_{ijk} \geq l_j \cdot I_{ijk}, \forall i,j,k \quad (11)$$

Minimum Skill Requirement

As explained before, in this model there is a minimum requirement for the skill an operator must have in order for him/her to be assigned to a cell station. This is put into a constraint through a comparison of the operator's actual skill and the skill that is required at the cell station. The way this constraint works is by stating that the operator's assignment skill must be greater than or equal to the product of the operator's assignment required skill multiplied by the binary assignment flag I_{ijk}. Since I_{ijk} will equal one when there is an assignment, it is assured that the skill will be greater than or equal to the skill required. On the contrary, when there is no assignment the constraint will state that the operator skill has to be greater than or equal to zero.

The skill of an operator assigned to a cell station, must be greater than or equal to the minimum skill required to work at that same cell station.

$$\mathrm{s_{ijk}} \geq \mathrm{r_{ijk}} \cdot \mathrm{I_{ijk}}\ , \forall\ i, j, k \tag{12}$$

Operator's Cell Training

In order for an operator to be assigned inside a cell, it must have prior training in that cell. That training is represented by the parameter t_{ij}, which is a binary variable that has a value of 1 when the operator *i* has training at the cell *j*, and 0 otherwise. This constraint was created by comparing the training parameter, with the assignment. The assignment has to be less or equal than the training in order to satisfy the constraint. By analyzing the constraint it is seen that if the training is 0, the assignment is forced to be 0. On the contrary, if the training has a value of 1 the assignment can acquire a value greater than zero.

An operator cannot be assigned to a cell if he is not trained to work at it.

$$A_{ijk} \leq t_{ij} \ , \forall \ i, j, k \tag{13}$$

Operator Total Maximum Assignment

As explained before, cellular manufacturing systems lend itself to employee cross-training, allowing for more assignment flexibility at the same time. Flexibility makes the demand of operators in each cell to be less than the actual amount of stations in it. This difference between operators' demand and the quantity of stations is due to operators been trained to work at more than one station inside a cell, consequently assigning these to do multiple tasks. This constraint states that the sum of all the assignments to the different cell stations for each operator must be less than or equal to one. Meaning than an operator cannot work more than one work shift or 100% of the available time.

The total assignment of an operator must be less than or equal to one.

$$\sum_{j=1}^{c} \sum_{k=1}^{e_j} A_{ijk} \leq 1 \ , \forall \ i \tag{14}$$

Non negative Assignments

One basic constraint introduced next is that the assignment decision variable, which represents the assignments of the operators, cannot acquire negative values.

An assignment must be greater than or equal to zero.

$$A_{ijk} \geq 0, \ \ \forall\, i, j, k \tag{15}$$

Cell Movement Constraint

With the purpose of giving the user a more realistic model that covers different manufacturing environments, a group of constraints was created so that the user could specify a maximum number of cells that an operator can be assigned to. In case the user wishes to follow an intracell movement policy, it would only need to set that maximum to one. In order to create this constraint, the binary decision variable Y_{ij} was introduced in equation 16, acquiring a value of 1 if the operator *i* is allowed to be assigned to the cell *j*, and 0 otherwise. Once Y_{ij} was introduced, two constraints were created to reach the goal of limiting the assignments of an operator to the cell limit imposed. Equation 17 states that the sum of Y_{ij} for each operator *i*, must equal ***v***. This means that each operator will be allowed to be assigned to the number of cell specified by ***v***, which represents the maximum amount of cells allowed by operator. Our real goal here is to make the assignment variable A_{ijk} to reflect this situation. For this matter equation 18 was created relating A_{ijk} and Y_{ij}. The constraint states that for each operator and cell the sum of assignments to all the stations must be less or equal to Y_{ij}. We can see here that for each operator there can be an assignment only in the station cells allowed by Y_{ij}.

An operator cannot be assigned to more than the maximum number of cells allowed.

$$Y_{ij} \in \{1,0\}\,, \quad \forall\, i, j \tag{16}$$

$$\sum_{j=1}^{c} Y_{ij} \leq v\,, \forall\, i \tag{17}$$

$$\sum_{k=1}^{e_j} A_{ijk} \leq Y_{ij}\,, \forall\, i, j \tag{18}$$

Stations Movement Constraint

This constraint provides the user with the ability to set a maximum number of stations that an operator can be assigned to. In order to create this constraint, the assignment binary flag I_{ijk} was compared with the parameter $\boldsymbol{h}$. This constraint states that for each operator, the sum of all the assignments can't be greater than $\boldsymbol{h}$.

An operator cannot be assigned to more than the maximum number of stations allowed.

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} I_{ijk} \leq h\,, \forall\, i \tag{19}$$

Bottleneck Assignments

An important detail of this model is to have the ability to assign the most skilled operators to the bottleneck stations. In order for the bottleneck objective function to achieve its goal there were some constraints that needed to be modified, and others that needed to be added to the model. The constraint that had to be modified was the operator time requirement constraint. Equation 20 shows that the modification done was to multiply the time requirement by the bottleneck parameter b_{jk}. By doing this we assure that there will be assignment only in the cell stations that are bottlenecks, and that the total assignment in these stations equals the requirement in the bottleneck cell station.

$$\sum_{i=1}^{w} BA_{ijk} = b_{jk} \cdot d_{jk} \text{ ,} \forall\, j, k \tag{20}$$

Two constraints were created in order to fulfill the necessity of having a bottleneck station with a dedicated operator. Equation 21 is the first constraint and ensures that each bottleneck station gets assigned no more than one operator. This is done by stating that the number of assignments done for each bottleneck has to equal the bottleneck parameter b_{jk}, which is one or zero, as shown in the following equation.

$$\sum_{i=1}^{w} I_{ijk} = b_{jk} \text{ ,} \forall\, j, k \tag{21}$$

The second constraint is equation 21 and ensures the attention in the bottleneck stations by requiring that each operator is assigned to no more than one bottleneck station. This constraint is created by stating that the maximum number of assignments for each operator has to be less than or equal to one.

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} I_{ijk} \leq 1 \text{ , } \forall\ i \tag{22}$$

Equation 23 is the last constraint created, and is introduced in order to relate the bottleneck assignment to the final resulting assignment A_{ijk}. The bottleneck assignments BA_{ijk} are an input to the sub model that will make the final assignment A_{ijk}. In order to create this constraint a new assignment variable NA_{ijk} was introduced so that the sum of the bottleneck assignment plus the new assignments added must equal the final assignment decision variable.

$$A_{ijk} = BA_{ijk} + NA_{ijk} \text{ , } \forall\ i, j, k \tag{23}$$

It is important to mention that all of the same constraints previously introduced apply for the non-bottleneck model, with the difference that the final assignment decision variable A_{ijk} is interchanged by the bottleneck assignment decision variable BA_{ijk}. These constraints include Assignment Binary Flag Control, Operator Cell Training, Operator Total Maximum Assignment, Operator Minimum Assignment, Minimum Skill Requirement, and the Non Negative Assignment constraint. It's important to mention that the Station and Cell movement constraints are not used in the bottleneck model because the extra constraints added in the bottleneck model cover these functionalities.

Cross-training Constraints

The following constraints are used in the cross-training model. These were introduced in the model with the purpose of making the skill levels increase go in accordance to the model's way of functioning, and in order to document the new skill levels. Equation 24 and equation 25 state that the cell training CT_{ij}, and the new cell training NT_{ij} are binary variables. Equation 26 and equation 27 deal with the cell training levels. Equation 26 states that the new cell training NT_{ij} is the sum of the actual training t_{ij} plus the cell training given CT_{ij}, and equation 27 states that actual training t_{ij} plus the cell training given CT_{ij} has to be less than or equal to one. This is because one of the two variables has to equal one; if an operator already has cell training it cannot receive more cell training.

$$NT_{ij} \in \{1,0\}, \forall\, i,j \tag{24}$$

$$CT_{ij} \in \{1,0\}, \forall\, i,j \tag{25}$$

$$t_{ij} + CT_{ij} = NT_{ij}, \forall\, i,j \tag{26}$$

$$t_{ij} + CT_{ij} \leq 1, \forall\, i,j \tag{27}$$

The next set of constraints deal with skill training awarded to the operators in the different cell stations. Equation 28 and equation 29 state that the skill training ST_{ijk}, and the new skill level NS_{ijk} are integer variables. Equation 30 and equation 31deal with the skill training levels. Equation 30 states that the new skill NS_{ijk} is the sum of the actual skill s_{ijk}, plus the skill training given ST_{ijk} and the cell training given CT_{ij}. Equation 31 states that the new skill training NS_{ijk} has to be less than or equal to ten, which is the greatest skill training possible in the model.

$$ST_{ijk} \in \{Int\}, \forall\, i, j, k \tag{28}$$

$$NS_{ijk} \in \{Int\}, \forall\, i, j, k \tag{29}$$

$$s_{ijk} + ST_{ijk} + CT_{ij} = NS_{ijk}, \forall\, i, j, k \tag{30}$$

$$s_{ijk} + ST_{ijk} + CT_{ij} \leq 10, \forall\, i, j, k \tag{31}$$

The next constraints are created in order to make the model to fully train an operator in a certain cell station. A skill training of one to an operator in a specific cell station does not do any improvement if the operator doesn't meet the minimum skill requirements imposed in the cell station. For that reason these constraints assure that if an operator receives any skill training, this training has to lead the operator to be on the minimum skill requirements of the cell station. The first step to achieve this target was to identify through a variable if there was skill training made. The variable Q_{ijk} was introduced to perform this duty, where if there's no training, Q_{ijk} equals zero, and equals one if ST_{ijk} acquire values between one and nine.

$$Q_{ijk} = \begin{cases} 0 \; for \; ST_{ijk} = 0 \\ 1 \; for \; 1 \leq ST_{ijk} \leq 9 \end{cases} \tag{32}$$

The following set of three constraints is used to create the previous shown desired properties.

$$Q_{ijk} \in \{1,0\}, \forall\, i,j,k \tag{33}$$

$$ST_{ijk} \geq Q_{ijk}, \forall\, i,j,k \tag{34}$$

$$ST_{ijk} \leq 9 \cdot Q_{ijk}, \forall\, i,j,k \tag{35}$$

Once Q_{ijk} is introduced we proceeded to create the constraints for two different scenarios. The first scenario is when an operator receives cell training in a cell it didn't have training before. When an operator receives training in a new cell it starts with a skill level of one in every station of the cell. In this case it has to be assured that after the cell training is received the operator must receive skill training in order to achieve the minimum skill requirement in at least one of the stations in the cell. In equation 36 can be seen that this constraint forces a limit in the skill training only when there has been cell training with additional skill training. This means that the minimum requirement is only imposed when Q_{ijk} and CT_{ij} both equal one. All other possible scenarios make the constraint to be satisfied without forcing the system to meet the minimum skill requirements.

$$CT_{ij} + ST_{ijk} \geq r_{jk} \cdot Q_{ijk} - M \cdot \left(1 - CT_{ij}\right), \forall\, i,j,k \tag{36}$$

The second scenario is when an operator already has cell training and receives skill training to increase the skill level to the minimum skill requirement. Like in the previous scenario it has to be assured that if an operator receives skill training, its level has to be greater than or equal to the minimum skill requirement. It can be seen in equation 37 that it limits the skill training to the minimum requirement only when there was previously cell training, and when there is an increase in skill level. This means that the minimum requirement is only imposed when Q_{ijk} equals one, and when CT_{ij} equals zero. All other possible scenarios make the constraint to be satisfied without forcing the system to meet the minimum skill requirements.

$$s_{ijk} + ST_{ijk} \geq r_{jk} \cdot \left(1 - CT_{ij}\right) - M \cdot \left(1 - Q_{ijk}\right), \forall\, i, j, k \tag{37}$$

The next two constraints are created in order to complement the two previous scenarios introduced. Equation 38 states that if an operator has no training in a cell, there cannot be any skill training on any station on that cell for that operator. On the contrary, if an operator has training in a cell he could have skill training in any of the stations in that cell. Equation 39 states that if an operator was given training in a cell, he must have skill training in at least one of the stations in that cell. On the contrary, if the operator didn't have any new cell training, it isn't required to have skill training in any of the stations of that cell.

$$NT_{ij} \geq Q_{ijk}, \forall\, i, j, k \tag{38}$$

$$\sum_{k=1}^{e_j} Q_{ijk} \geq CT_{ij}, \forall\, i, j \tag{39}$$

The last constraint to be introduced imposes a budget to the cross-training to be given in the system. This budget limits the cross-training and pushes the model to cross-train specifically in the places where the objective function will be better attended. The constraint is created by stating that the cross-training cost in the system has to be less than or equal to the available budget for cross-training. The cross-training cost is composed of two costs; the cell training cost, which is the cost of training operators to work in a new cell they didn't have training in, and the skill training cost, which is the cost of increasing the skill level of operators in a specific cell station.

$$\sum_{i=1}^{w}\sum_{j=1}^{c} c_j CT_{ij} + \sum_{i=1}^{w}\sum_{j=1}^{c}\sum_{k=1}^{e_j} sc_{jk} ST_{ijk} \leq B \tag{40}$$

3.5 Models

The next four sections present the linear optimization model created for this work. In fact, there are four sub models that are executed continuously, one after the other, in order to achieve the desired results. The four sub models are; the Bottleneck Stations model, the Minimum Skill-Satisfaction model, the Total System Skill-Satisfaction model, and at last the cross-training model.

3.5.1 Bottleneck Stations Model

$$Max \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} BA_{ijk}(s_{ijk} + w \cdot p_{ijk})$$

St.

$$\sum_{i=1}^{w} BA_{ijk} = b_{jk} \cdot d_{jk} \text{ ,} \forall\, j, k$$

$$\sum_{j=1}^{c} \sum_{k=1}^{e_j} BA_{ijk} \leq 1 \text{ ,} \forall\, i$$

$$I_{ijk} \in \{1{,}0\} \text{ ,} \forall\, i, j, k$$

$$BA_{ijk} \geq 0, \ \forall\, i, j, k$$

$$I_{ijk} \geq BA_{ijk} \text{ ,} \forall\, i, j, k$$

$$\sum_{i=1}^{w} I_{ijk} = b_{jk} \text{ ,} \forall\, j, k$$

$$BA_{ijk} \geq l_j \cdot I_{ijk} \text{ ,} \forall\, i, j, k$$

$$s_{ijk} \geq r_{ijk} \cdot I_{ijk} \text{ ,} \forall\, i, j, k$$

$$\sum_{j=1}^{c} \sum_{k=1}^{e_j} I_{ijk} \leq 1 \text{ ,} \forall\, i$$

$$BA_{ijk} \leq t_{ij} \text{ ,} \forall\, i, j, k$$

Figure 7. Bottleneck stations model

3.5.2 Minimum Skill-Satisfaction Model

$$Max\left\{Min\sum_{j=1}^{c}\sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk},\ \ \forall\, i\right\} = Max\,(U)$$

St.

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk} \geq U,\ \ \forall\, i$$

$$\sum_{i=1}^{w} A_{ijk} = d_{jk}\,, \forall\, j,k$$

$$I_{ijk} \in \{1,0\}\,, \forall\, i,j,k$$

$$I_{ijk} \geq A_{ijk}\,, \forall\, i,j,k$$

$$A_{ijk} \geq l_j \cdot I_{ijk}\,, \forall\, i,j,k$$

$$\mathrm{s_{ijk}} \geq \mathrm{r_{ijk}} \cdot \mathrm{I_{ijk}}\,, \forall\, i,j,k$$

$$A_{ijk} \leq t_{ij}\ \ , \forall\, i,j,k$$

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} A_{ijk} \leq 1\,, \forall\, i$$

$$A_{ijk} \geq 0,\ \ \forall\, i,j,k$$

$$Y_{ij} \in \{1,0\}\,,\ \ \forall\, i,j$$

$$\sum_{j=1}^{c} Y_{ij} \leq v\,, \forall\, i$$

$$\sum_{k=1}^{e_j} A_{ijk} \leq Y_{ij}\,, \forall\, i,j$$

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} I_{ijk} \leq h\,, \forall\, i$$

$$A_{ijk} = BA_{ijk} + NA_{ijk}\ \ , \forall\, i,j,k$$

Figure 8. Minimum operators' skill-satisfaction model

3.5.3 Total System Skill-Satisfaction Model

$$Max \sum_{i=1}^{w}\sum_{j=1}^{c}\sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk}$$

St.

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} p_{ijk}\, s_{ijk} A_{ijk} \geq U, \ \ \forall\, i$$

$$A_{ijk} \geq \ 0, \ \ \forall\, i, j, k$$

$$Y_{ij} \in \ \{1{,}0\}\,, \ \ \forall\, i, j$$

$$\sum_{i=1}^{w} A_{ijk} = d_{jk}\,, \forall\, j, k$$

$$\sum_{j=1}^{c} Y_{ij} \leq v\,, \forall\, i$$

$$I_{ijk} \in \ \{1{,}0\}\,, \forall\, i, j, k$$

$$I_{ijk} \geq \ A_{ijk}\,, \forall\, i, j, k$$

$$\sum_{k=1}^{e_j} A_{ijk} \leq Y_{ij}\,, \forall\, i, j$$

$$A_{ijk} \geq \ l_j \cdot I_{ijk}\,, \forall\, i, j, k$$

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} I_{ijk} \leq h\,, \forall\, i$$

$$\mathrm{s_{ijk}} \geq \mathrm{r_{ijk}} \cdot \mathrm{I_{ijk}}\,, \forall\, i, j, k$$

$$A_{ijk} \leq t_{ij} \ \ , \forall\, i, j, k$$

$$A_{ijk} = BA_{ijk} + NA_{ijk} \ \ , \forall\, i, j, k$$

$$\sum_{j=1}^{c}\sum_{k=1}^{e_j} A_{ijk} \leq 1\,, \forall\, i$$

Figure 9. Total system skill-satisfaction model

3.5.4 Cross-training Recommendations Model

$$Max \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} p_{ijk} \, NS_{ijk}$$

St.

$$NT_{ij} \in \{1,0\}, \forall\, i,j$$

$$CT_{ij} \in \{1,0\}, \forall\, i,j$$

$$t_{ij} + CT_{ij} = NT_{ij}, \forall\, i,j$$

$$t_{ij} + CT_{ij} \leq 1, \forall\, i,j$$

$$ST_{ijk} \in \{Int\}, \forall\, i,j,k$$

$$NS_{ijk} \in \{Int\}, \forall\, i,j,k$$

$$s_{ijk} + ST_{ijk} + CT_{ij} = NS_{ijk}, \forall\, i,j,k$$

$$s_{ijk} + ST_{ijk} + CT_{ij} \leq 10, \forall\, i,j,k$$

$$Q_{ijk} \in \{1,0\}, \forall\, i,j,k$$

$$ST_{ijk} \geq Q_{ijk}, \forall\, i,j,k$$

$$ST_{ijk} \leq 9 \cdot Q_{ijk}, \forall\, i,j,k$$

$$CT_{ij} + ST_{ijk} \geq r_{jk} \cdot Q_{ijk} - M \cdot \left(1 - CT_{ij}\right), \forall\, i,j,k$$

$$s_{ijk} + ST_{ijk} \geq r_{jk} \cdot \left(1 - CT_{ij}\right) - M \cdot \left(1 - Q_{ijk}\right), \forall\, i,j,k$$

$$NT_{ij} \geq Q_{ijk}, \forall\, i,j,k$$

$$\sum_{k=1}^{e_j} Q_{ijk} \geq CT_{ij}, \forall\, i,j$$

$$\sum_{i=1}^{w} \sum_{j=1}^{c} c_j CT_{ij} + \sum_{i=1}^{w} \sum_{j=1}^{c} \sum_{k=1}^{e_j} sc_{jk} ST_{ijk} \leq B$$

Figure 10. Cross-training recommendations model

3.6 Model's Guided User Interface

With the purpose of making the user experience as comfortable as possible, a guided user interface was created. The guided user interface allows the user to avoid direct interaction with Lingo11, which is the optimization software used to solve the model. Through the user interface created, the user can create new problems, input and export the model parameters, evaluate and export the model results, and most important of all execute the model, all without physically opening Lingo. The interface was created in Access 2007 using Forms, Tables, Queries, Macros, and Visual Basic for Applications (VBA) code.

Once the application is opened, the first screen shows a Main Menu with five different buttons. The first is the CREATE button, which is used to create a new case for the model. Another is the SET button, which is used in order to set the parameters the model uses to run. The RESULTS button opens a new window with the results provided by the model. The RUN button activates the connection between LINGO and ACESSS in order to execute the model and obtain the results, and last but not least, the EXIT APPLICATION button quits completely the interface application. Next, the actions of each button will be discussed in more detail so that all the functions of the user interface application are understood. Figure 11 shows the Main Menu of the application.

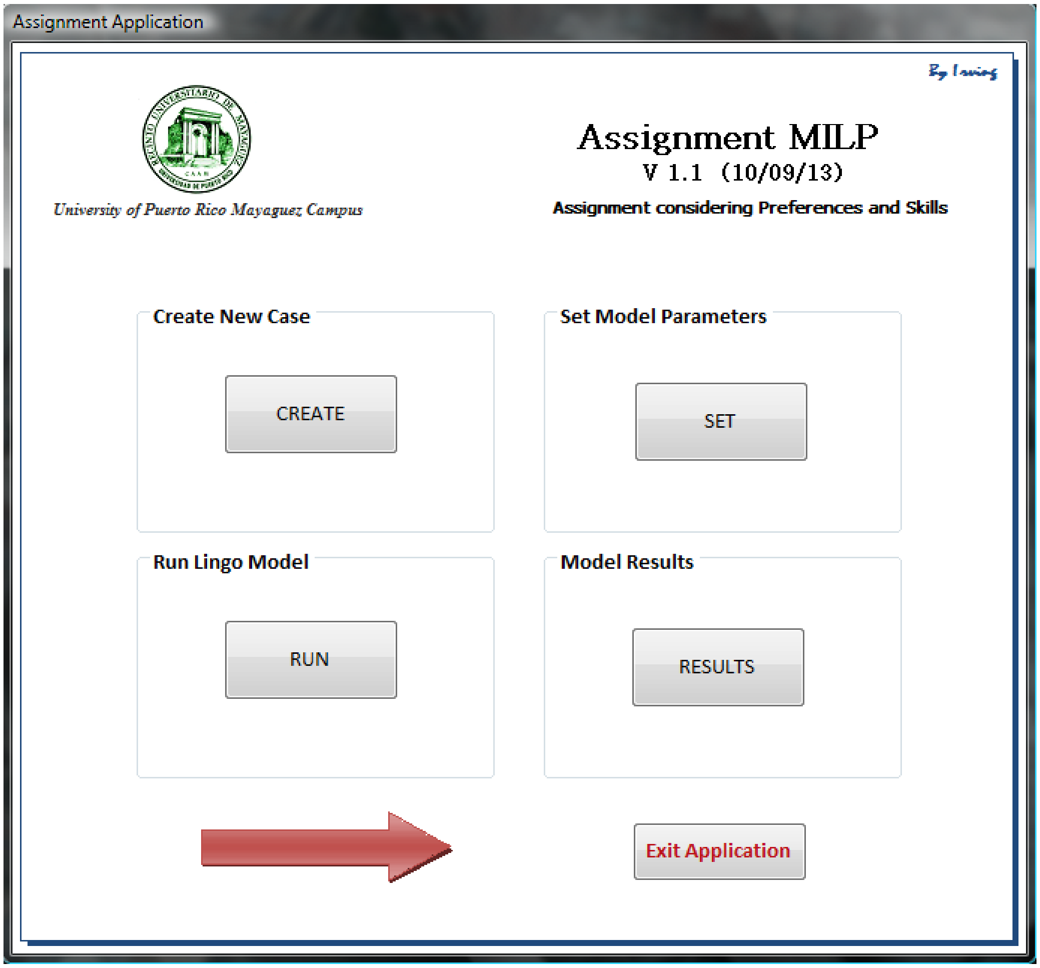

Figure 11. Interface main window

3.6.1 CREATE Button

The CREATE button takes you to two windows that help you in the process of creating a new case for the model. In order to create a new case there are certain parameters that have to be know in advance so that all the required parameters matrixes contain the appropriate dimensions. The first of those parameters are the numbers of operators and the number of cells in the system. These parameters are the first ones to be entered in the first window opened.

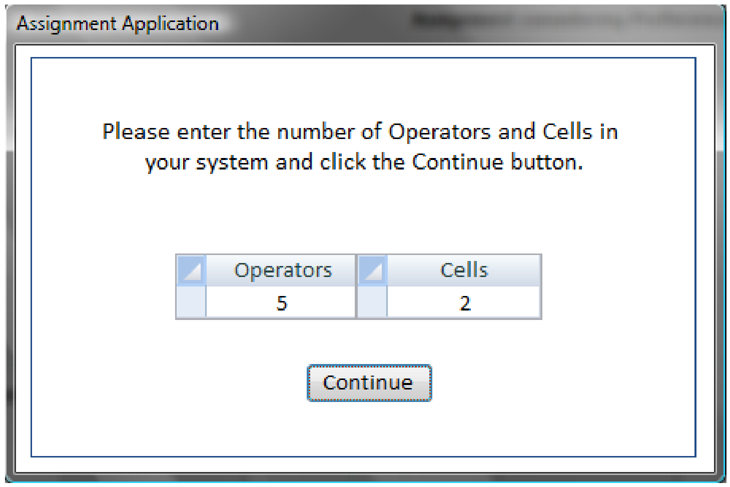

Figure 12. Interface new case window 1

In the first window the application will ask you to enter the amount of operators and cells in the system. After entering these two parameters the user has to click the CONTINUE button in order to proceed to the next window. The second window consists of the user entering the number of stations each cell contains. This data is entered in a second window because it depends on the number of cells in the system, which was entered first.

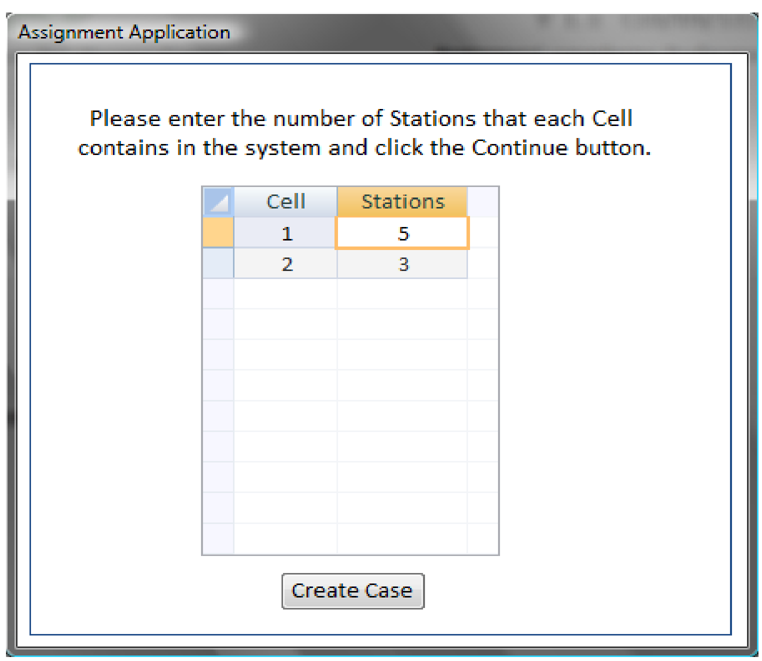

Figure 13. Interface new case window 2

Once the number of stations for each cell is introduced, the user has to click the CREATE CASE button. Automatically once the button is pressed the new case will be created and all the tables in the application will have the appropriate dimensions. The next step would be to set the parameter values.

3.6.2 SET Button

The SET button opens a new window that eases the functionality to set all the parameters related to the model. This window has a series of buttons, where each of them provides access to different model parameters. These buttons provide access to the following parameters: Cell Training, Minimum Assignment, Cell Station Parameters where you find the parameters relating to the different cell stations in the system, Operator Assignment Parameters where you find the Skills and Preferences of each operator in the different cell stations, Assignment Policy where you set the parameters relating to the intracell and intercell movement policies, and at last the cross-training parameters.

Besides the buttons to edit the model parameters there is a button in the right lower part of the window that says Export Parameters. That button takes all the parameter values already entered and exports them into a new created Excel file. This is a useful tool because it provides with the ability to easily store a case with the intention of using it again in the future. For example, any user could get the stored exported file, copy it, and then paste it in to the parameter tables. That way any old case could be easily set again in order to adjust some parameters and evaluate again results.

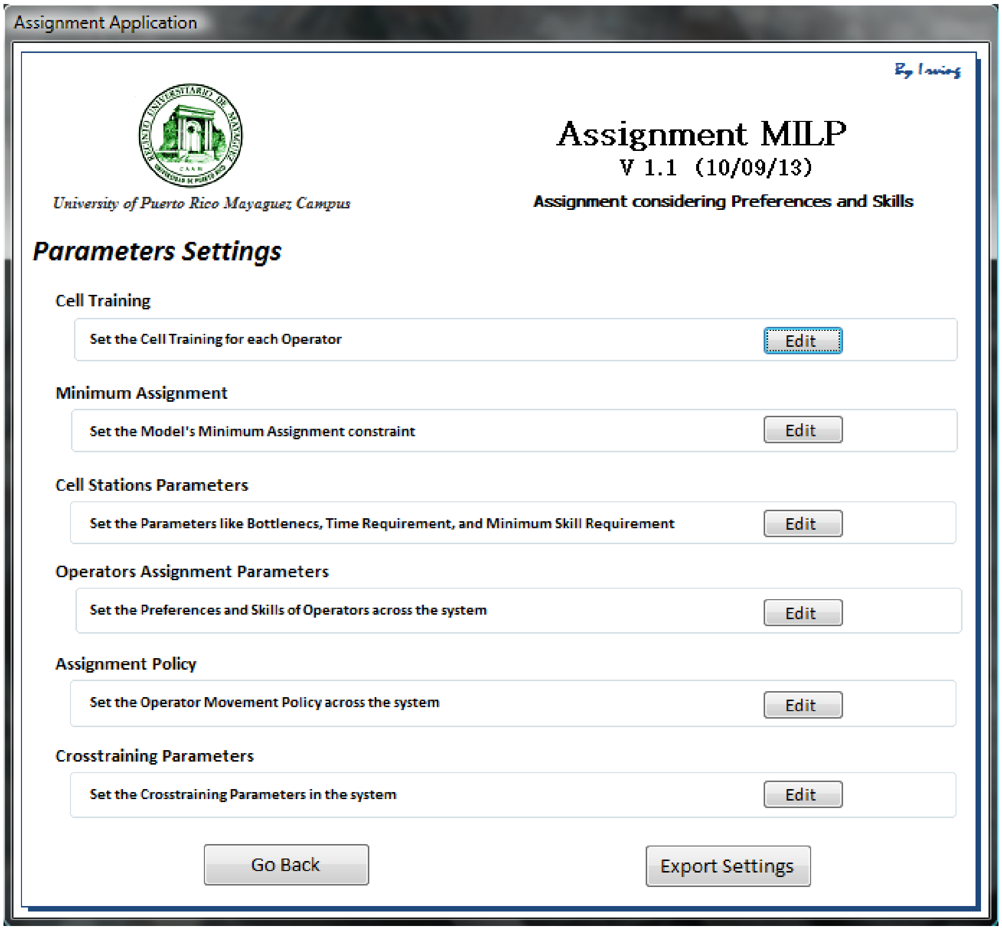

Figure 14. Interface parameters settings window

Each window opened by the EDIT buttons provides tables to input or edit the parameters values, and also boxes to filter the tables and enter the values in order, let's say by cell. These filters could be applied either by operator, cell, or station. Figure 15 shows as example the window that sets the Cross-training Parameters, where the user can decide whether to receive cross-training recommendations or not, the available budget for cross-training, and the related costs to the cross-training model, which are the cost to train an operator in a new cell, and the cost to increase the skill level of an operator in a specific cell station.

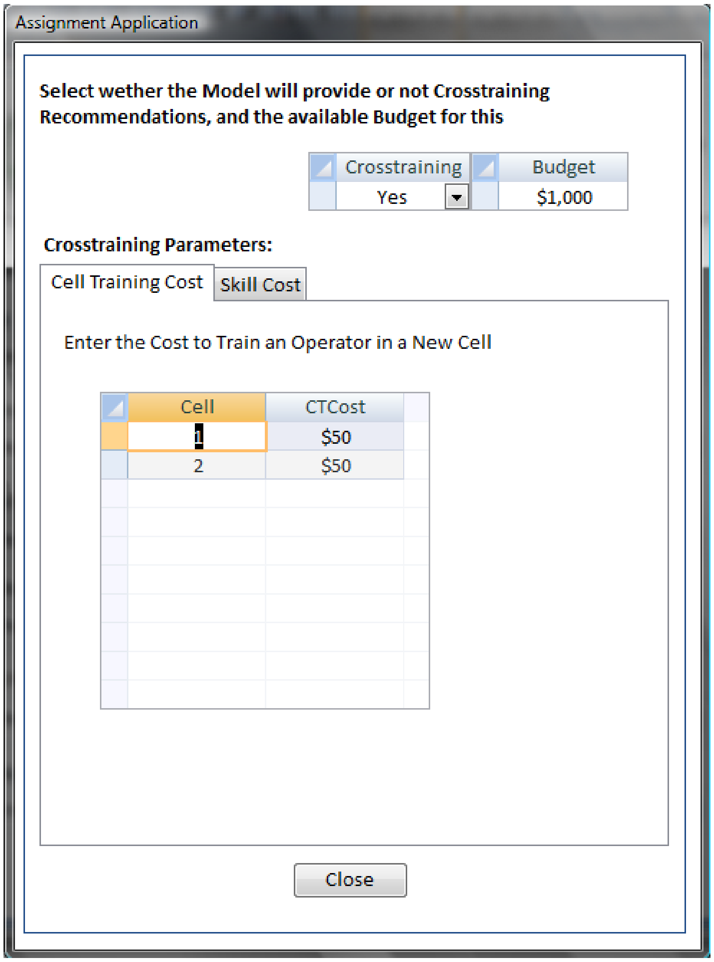

Figure 15. Interface cross-training settings window

3.6.3 RESULTS Button

The RESULTS button opens a new window that lets you review the results provided by LINGO. This window has a series of buttons, where each of them provides access to different model results for evaluation. The results are divided as results without cross-training, results with cross-training, and a comparison between the cross-training and no cross-training results.

Besides the buttons to open the model results, there is a button on the right lower part of the window that says Export Results, as in the Settings window. That button takes all the Results and exports them to a new created Excel file. This is a useful tool because it provides with the ability to easily store the results with the intention of analyzing them again in the future. For example, any user could get the stored exported file in order to compare them to a new case being currently evaluated, making it possible to compare old case results with a new case, and make decisions efficiently.

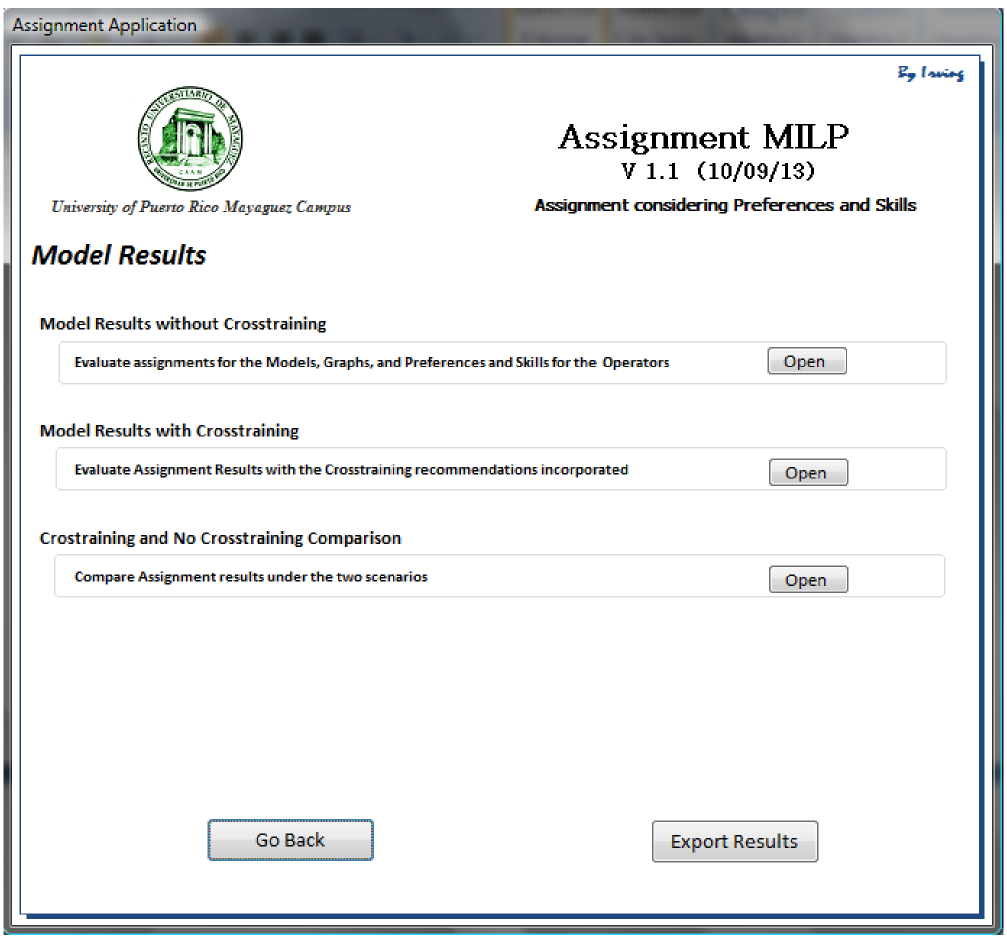

Figure 16. Interface results window

Each window opened by the OPEN buttons provides access to tables and graphs of the selected results. Figure 17 shows as example the window with the cross-training recommendations results, where the user can select the different tabs to see Bottleneck Assignments, Minimum Objective Function Assignments, Total Skill-Satisfaction Objective Function Assignments, Cross-training performed in the system, a Skill-Satisfaction Graph, and a graph with the preferences and skills of each operator in each cell station.

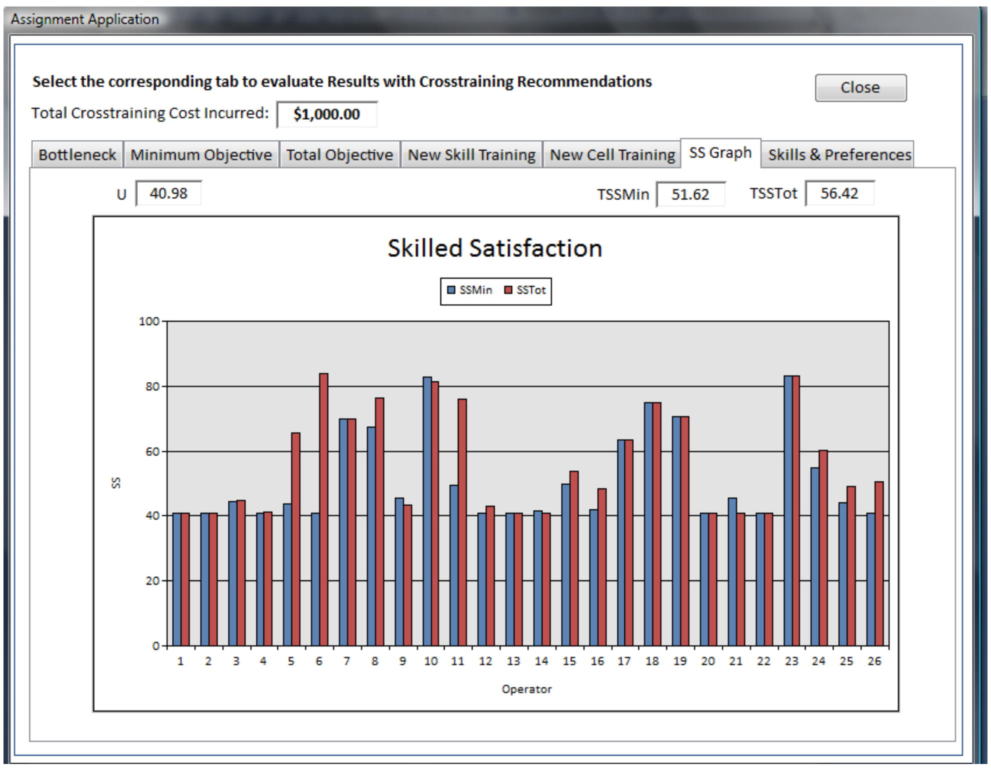

Figure 17. Interface results with cross-training window

3.6.4 RUN Button

The RUN Button activates the Visual Basic code that calls LINGO and connects it to Access in order to execute the model with the current parameters in the interface application. Once the model finishes executing you will receive a message indicating that the model executed successfully, and all the results will be available in the user interface.

3.6.5 EXIT APPLICATION Button

The EXIT APPLICATION Button closes the interface application completely. The only way to exit the application is through this button since no close button is available on the windows.

Chapter 4

This chapter discusses a hypothetical case created with the intention of evaluating the developed assignment linear models. All the parameters were intentionally set to test all the model functionalities and do not represent data information from any company. The case is composed of 5 cells with a variable number of stations by cell, 8-hour shifts, $1000 cross-training budget, and 23/26 operators depending on whether the problem is an intercell/intracell problem. The case created puts to the test all the constraints incorporated in the model with special attention to the bottleneck stations constraints, and intracell and intercell assignment policies constraints. A capacity analysis made for the case is discussed, followed by the introduction of the model parameters, and at last the results for the intracell and intercell assignment policies with its corresponding cross-training recommendations.

4.1 Case Study Capacity Analysis

An important parameter in this model is the number of operator available in the system. The number of operators needed should be calculated through a capacity analysis that depends primarily on the stations' standard time, and the demand units imposed on each cell. It is important to consider the amount of operators obtained by the capacity analysis, since setting more operators than needed could results in having significant idle time in the system for the operators. The capacity analysis is introduced first in order to continue with the model case parameters. First, the demand for each cell is shown in Table 1.

Table 1. Case study units demand per cell

Cell	Demand (units)
1	85
2	115
3	125
4	95
5	120

Using the station's standard times and total shift time, the requirement or shift percent required for each station in order to comply with the demand units for the cell can be calculated through the following formula in equation 41.

$$Shift\ \% = \frac{Station\ Required\ Time}{Total\ Available\ Time} = \frac{Cell\ Demand\ Units * ST}{Total\ Shift\ time} \tag{41}$$

It is important to mention that the standard times used in this model should be the result of a time study conducted in the selected manufacturing environment. A time study offers standard times that include allowances to account for the many interruptions, delays, and slowdowns cause by fatigue in every work assignment. Allowances normally adjudicated to workers in this environment are fatigue, personal needs, workstation cleaning, and unavoidable delays (Niebel & Freivalds, 2003). Due to the fact that we use standard times obtained from a time study, if an operator is assigned a complete shift to a cell station ($A_{ijk} = 1$) it is understood that he/she will have the proper allowances included in his/her shift, thus it will have breaks included for personal needs, such as going to the restroom. For matter of illustration, Table 2 illustrates the standard times and fractions of shift required for each station in cell 2, calculated using the previous formula and the demand of 115 units in cell 2.

Table 2. Case study standard time and shift % required per station for cell 2

Cell	Station	1	2	3	4
2	ST (min/unit)	3.1	3.4	2.7	2.5
	Shift fraction	0.74	0.81	0.65	0.60

Since the production demand for each cell has to be achieved, in order to comply with the demand in only one shift the number of operators has to be greater than or equal to the shift fraction needed for complying demand in a cell. For example, if the capacity analysis says that there is needed a 2.80 shift fraction for meeting production demand in cell 2, then three operators are required in cell number 2 for meeting the required demand in the cell. Table 3 shows the amount of operators required for each cell after doing the previous calculation for every cell.

Table 3. Operators required per cell

	Intercell	Intracell
Cell	Operators required	
1	3.33	4
2	2.80	3
3	6.12	7
4	5.13	6
5	5.43	6
Operators	**22.80 (23)**	**26**

We can see through the capacity analysis that there are a total of 26 operators needed in the system to comply with the demand. It is important to understand that the total number of operators needed in the system is not always the sum of the operators needed in each cell. This will depend on the assignment policy used in the system; intracell or intercell assignment. When there is an intracell assignment policy each cell is treated separately as an individual system and the total number of operators needed to meet demand is the sum of the operators required for each cell. On the other hand, when there is an intercell assignment policy the total number of operators is calculated through the total shift fraction in the whole system. For example, in this case the total shift fraction in the system is 22.80, hence there are 23 operators needed for an intercell policy. On the other hand, if the case presented behaves as an intracell policy there would be needed 26 operators, which is the sum of the operators needed in each cell. This happens because in intracell assignments, operators are confined to only one cell, leaving no opportunity to assign any operator with free time to another cell.

4.2 Case Study Parameters

As a matter of illustration a sample data pertaining to skills, preferences, trainings, time requirements, and bottlenecks, etc., is now presented for cell 2. The minimum assignment possible in cell 2 is .0625 which is equal to 30 minutes in an 8 hour shift, the cost for training a new operator in cell 2 is $53, and the cross-training budget is $1,000. For the complete case study data please refer to the Appendix D.

Table 4. Case study preferences matrix for Cell 2

Preference	Cell 2				
Operators\Stations	1	**2**	3	4	Cell Training
1	6	7	8	10	**X**
2	7	2	6	5	**X**
3	5	5	10	1	**X**
4	6	6	5	6	**X**
5	8	1	8	10	
6	4	2	10	9	**X**
7	6	4	6	7	
8	4	2	2	8	**X**
9	3	3	9	4	
10	1	10	5	6	**X**

Preference	Cell 2				
Operators\Stations	1	**2**	3	4	Cell Training
11	10	3	5	1	
12	6	10	1	6	
13	10	5	1	2	
14	5	4	3	6	**X**
15	3	9	8	3	**X**
16	3	8	5	5	
17	6	1	3	6	**X**
18	1	6	3	5	
19	7	7	4	8	**X**
20	1	8	6	7	**X**
21	2	1	1	2	**X**
22	9	4	7	4	
23	4	5	4	9	
24	8	7	4	10	
25	10	3	6	8	**X**
26	7	5	6	7	

Table 4 shows the preferences matrix for cell 2, where a red color scale preference indicates the preference level. Also it could be seen to the right an X showing if the operator has training or not in the cell. On the other hand, Table 5 shows a cross-training matrix for cell 2, where there is a red color background in the stations where the operator is not permitted to work due to minimum skill requirements, or need of training in the cell. Also are shown the time requirement, minimum skill, and skill training cost for each station.

Table 5. Case study cross-training matrix, time requirement, and skill cost for cell 2

Skill	Cell 2				
Operators\Stations	1	**2**	3	4	Cell Training
1	8	3	8	4	**X**
2	1	5	7	7	**X**

Skill	Cell 2				
Operators\Stations	1	**2**	3	4	Cell Training
3	6	7	8	9	**X**
4	5	5	7	8	**X**
5	0	0	0	0	
6	3	5	7	6	**X**
7	0	0	0	0	
8	7	5	5	7	**X**
9	0	0	0	0	
10	7	5	6	4	**X**
11	0	0	0	0	
12	0	0	0	0	
13	0	0	0	0	
14	6	8	1	7	**X**
15	1	8	8	6	**X**
16	0	0	0	0	
17	6	5	8	6	**X**
18	0	0	0	0	
19	6	4	7	5	**X**
20	3	3	7	5	**X**
21	2	8	5	6	**X**
22	0	0	0	0	
23	0	0	0	0	
24	0	0	0	0	
25	5	5	9	6	**X**
26	0	0	0	0	
Minimum Required Skill	3	5	4	6	
Time Requirement	0.74	**0.81**	0.65	0.60	
Skill Training Cost	$12	$8	$8	$14	

4.3 Results

The linear model was programmed in LINGO 11 and results for the two assignment policy scenarios, as well as the cross-training model recommendations are shown with the purpose of comparison. Moreover, with the intention of showing how the model works, results from individual objective function stages are shown.

4.3.1 Intercell Assignment Policy Results

For the intercell case composed of 23 operators, the results without the cross-training recommendations will be presented first, followed by the results achieved after implementing the cross-training recommendations.

4.3.1.1 Intercell No Cross-training Results

We first start by discussing the results of the bottleneck assignments. The bottleneck assignments are made by mainly considering the skills of the operators, the model tries to maximize the total sum of the skill assigned to the bottlenecks. Table 6 shows the bottleneck of each cell, along with its corresponding operator's assignment, preference and skill associated with it.

Table 6. Bottleneck stations assignment with skills and preferences for the intercell policy

Cell	Bottleneck Station	Operator	Assignment	Skill	Preference
1	4	20	0.83	7	10
2	2	15	0.81	8	9
3	8	17	0.99	8	8
4	1	19	0.79	8	10
5	1	7	1.00	7	10

By looking at Table 6 it can be seen that every operator assigned to a bottleneck station has a skill of eight at the bottleneck station except the operators in cell 1 and cell 5, which have a skill of seven. Inspecting cell 5 in more detail, it can be seen that, out of all the operators, the maximum skill on the bottleneck station of cell 5 is seven, which completely explains why an operator with a skill greater than seven in that station was not assigned. Figure 18 shows all the operators along with their associated skills in the bottleneck station of cell 5.

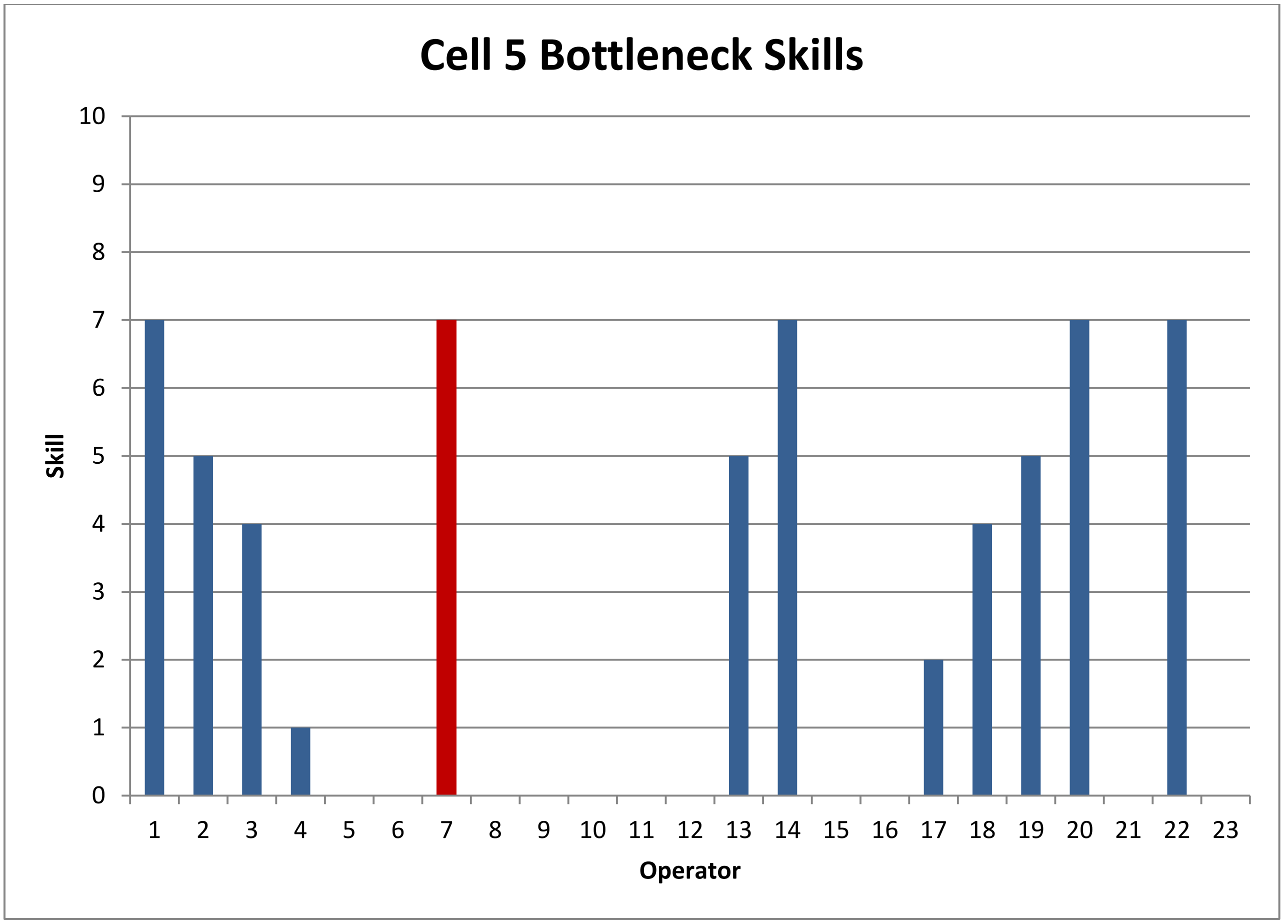

Figure 18. Bottleneck assignments for cell 5 of the intercell policy

Once the bottleneck assignments were made, the model proceeded to make the rest of the assignments by leaving the bottlenecks assignments fixed, and with the objective of maximizing the minimum skill-satisfaction between all the operators in the system. At last, once the minimum skill-satisfaction was maximized, the model proceeded to input that minimum as a constraint into the next stage, which is maximizing the total system skill-satisfaction. Figure 19 shows how the total system skill-satisfaction increased from the second stage to the third stage while leaving the minimum satisfaction equal.

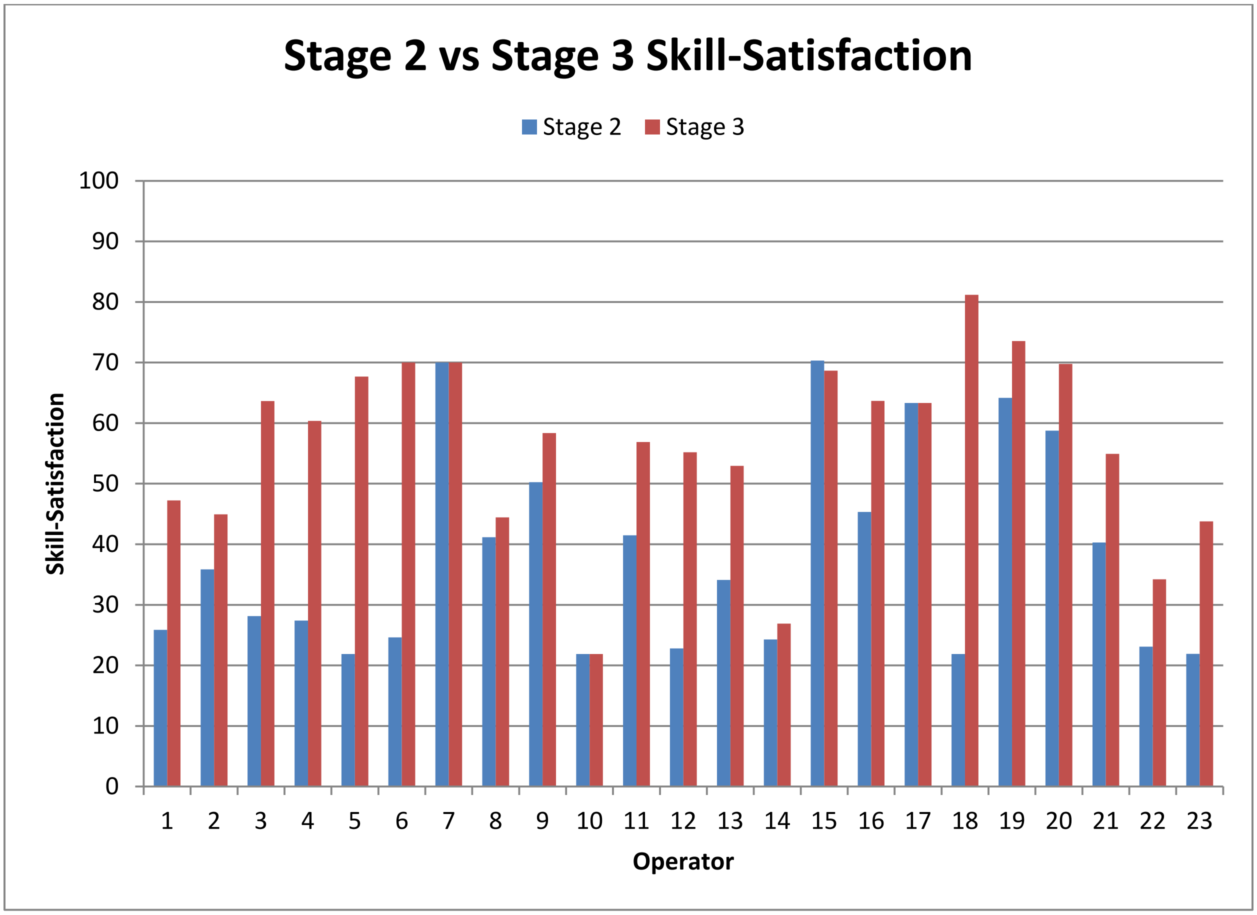

Figure 19. Skill-satisfaction by operator for stages 2 and 3 of the intercell policy

The increase in total system skill-satisfaction from 38.21 to 56.24 for the change in stages is clearly appreciated. Table 7 summarizes the numbers for these two stages, where it can be seen how, by applying the third stage; the total system skill-satisfaction was increased while leaving the minimum skill-satisfaction at the maximum possible value.

Table 7. Stages performance for the intercell policy

Stage	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction
2	21.88	38.21
3	21.88	56.24

Lastly, Table 8 shows the resulting none zero final assignments for the model. Along with the assignments, the skill and preference for the assignment cell station is presented. Both skill and preference values for the assignments are always maintained relatively high, mostly at values of five or higher.

Table 8. Preferences, skills, and final assignments for the intercell policy

Operator	Cell	Station	Preference	Skill	Final Assignment
1	1	6	7	8	0.31
1	3	7	6	5	0.27
1	4	7	10	7	0.07
1	5	3	8	6	0.35
2	1	5	9	7	0.06
2	3	7	6	8	0.27
2	5	2	7	6	0.67
3	1	2	10	7	0.35
3	4	6	8	8	0.34
3	5	3	8	5	0.06
3	5	5	10	6	0.24
4	1	1	9	7	0.27
4	3	3	8	6	0.25
4	4	3	7	7	0.11
4	4	8	10	7	0.38
5	3	3	9	7	0.48
5	3	5	9	8	0.52
6	4	2	10	7	0.63
6	4	7	10	7	0.37
7	5	1	10	7	1.00
8	2	1	4	7	0.39
8	2	4	8	7	0.60
9	1	3	10	7	0.58
9	1	6	6	7	0.42

Operator	Cell	Station	Preference	Skill	Final Assignment
10	2	1	1	7	0.35
10	2	3	5	6	0.65
11	3	1	10	5	0.55
11	3	4	10	7	0.39
11	3	7	5	7	0.06
12	4	3	9	6	0.49
12	4	4	6	7	0.18
12	4	6	8	8	0.33
13	5	4	10	6	0.41
13	5	7	8	6	0.59
14	5	2	1	6	0.13
14	5	6	7	4	0.77
14	5	7	8	6	0.09
15	2	2	9	8	0.81
15	4	4	9	6	0.19
16	3	2	9	8	0.09
16	3	9	10	6	0.65
16	4	7	10	7	0.26
17	3	8	8	8	0.99
18	3	1	8	7	0.26
18	3	2	10	9	0.74
19	4	1	10	8	0.79
19	5	3	7	7	0.21
20	1	4	10	7	0.83
20	3	1	8	7	0.11
20	5	4	10	9	0.06
21	1	5	9	6	0.50
21	3	6	8	7	0.49
22	5	5	7	6	0.58
22	5	7	5	8	0.25
23	4	4	4	8	0.31
23	4	5	7	7	0.69

4.3.1.2 Intercell with Cross-training Results

This section will present the bottleneck assignments, final assignments, and resulting skill training, along with the new minimum and total skill-satisfaction of the intercell system with the cross-training recommendations incorporated. Out of the 23 operators in the intercell policy system, the model provided skill training to 51 different operator cell and station combinations across the system, and 5 operators received training in new cells. The cross-training budget was taken to the limit, which means there was a cross-training cost of $1,000. Table 9 presents the operators that received training in a new cell.

Table 9. Operators receiving new cell training for the intercell policy

Operator	Cell
10	3
14	3
15	3
17	1
23	1

It can be seen that there was new cell training in cells 3 and 1. Table 10 presents the operator and cell station where skill training was given, along with the preference, previous skill, and resulting new skill level of the operator at the given cell station.

Table 10. Cross-training data along with preferences for the intercell policy

Operator	Cell	Station	Preference	Skill	Skill Training	New Skill
1	3	4	10	6	4	10
3	1	4	9	6	4	10
3	2	3	10	8	2	10
4	1	4	9	5	5	10
6	2	3	10	7	3	10
8	4	4	10	2	8	10
10	2	2	10	5	5	10
10	3	1	2	0	1	1
10	3	2	9	0	1	1
10	3	3	9	0	1	1
10	3	4	2	0	1	1
10	3	5	3	0	1	1
10	3	6	5	0	1	1
10	3	7	10	0	10	10

Operator	Cell	Station	Preference	Skill	Skill Training	New Skill
10	3	8	1	0	1	1
10	3	9	2	0	1	1
11	3	4	10	7	3	10
14	3	1	2	0	1	1
14	3	2	6	0	1	1
14	3	3	5	0	1	1
14	3	4	9	0	6	6
14	3	5	9	0	1	1
14	3	6	1	0	1	1
14	3	7	6	0	1	1
14	3	8	7	0	1	1
14	3	9	10	0	1	1
15	1	4	10	6	4	10
15	2	2	9	8	2	10
15	3	1	3	0	1	1
15	3	2	9	0	1	1
15	3	3	5	0	1	1
15	3	4	7	0	1	1
15	3	5	5	0	1	1
15	3	6	3	0	1	1
15	3	7	9	0	10	10
15	3	8	6	0	1	1
15	3	9	2	0	1	1
15	4	4	9	6	4	10
17	1	1	9	0	1	1
17	1	2	7	0	1	1
17	1	3	2	0	1	1
17	1	4	10	0	10	10
17	1	5	7	0	1	1
17	1	6	6	0	1	1
20	1	4	10	7	3	10
23	1	1	5	0	1	1
23	1	2	6	0	1	1
23	1	3	10	0	5	5
23	1	4	10	0	10	10
23	1	5	8	0	1	1
23	1	6	9	0	1	1

With regards to Table 9, it can be seen that all the skill training was performed in cell stations were the operators had high preferences. It's important to mention that the case when the operators had a skill of zero, and a skill training of one in a station, means that the operators only received the new cell training which increased their skill level to the minimum possible of one. Nonetheless, when an operator receives new cell training, they receive additional skill training in order to increase the skill level to the minimum required in the station in at least in one of the stations of the cell. This also explains why there are operators that received a minimum skill training of one, in a station with a low preference. Revising in more detail those cases it can be appreciated that the operators received a training conducting to the minimum skill required in the station where the operator had the highest preference.

Table 11 shows the new resulting bottleneck assignments after the cross-training recommendations. It can be seen that the operator assignments changed for cell 1 and 2. Originally in the bottleneck station of cell 1 the operator 20 was assigned, and after cross-training recommendations the operator 23 was selected. In cell 2 the operator 15 was originally assigned and changed to operator 10 after cross-training recommendations. These two changes in the assignments are due to the fact that these two operators now have the maximum skill of 10 due to the cross-training performed.

Table 11. Cross-training bottleneck assignments for the intercell policy

Cell	Bottleneck Station	Operator	Assignment	Skill	Preference
1	4	23	0.83	10	10
2	2	10	0.81	10	10
3	8	17	0.99	8	8
4	1	19	0.79	8	10
5	1	7	1.00	7	10

Table 12 shows the final assignments after implementing the cross-training recommendations provided by the model.

Table 12. Cross-training final assignments for the intercell policy

Operator	Cell	Station	Preference	Skill	Final Assignment
1	1	6	7	8	0.31
1	2	1	6	8	0.63
1	5	3	8	6	0.06
2	1	5	9	7	0.06
2	5	2	7	6	0.80
2	5	6	6	8	0.14
3	1	2	10	7	0.35
3	2	3	10	10	0.28
3	5	3	8	5	0.20
3	5	6	6	5	0.17
4	1	1	9	7	0.27
4	2	1	6	5	0.12
4	2	4	6	8	0.09
4	4	3	7	7	0.14
4	4	8	10	7	0.38
5	3	3	9	7	0.48
5	3	5	9	8	0.52
6	2	3	10	10	0.37
6	4	2	10	7	0.63
7	5	1	10	7	1.00
8	2	4	8	7	0.33
8	4	4	10	10	0.67
9	1	3	10	7	0.58
9	1	6	6	7	0.42
10	2	2	10	10	0.81
10	3	7	10	10	0.19
11	3	1	10	5	0.27
11	3	4	10	10	0.39
11	3	9	9	7	0.34
12	4	3	9	6	0.33
12	4	6	8	8	0.67
13	5	3	4	6	0.16
13	5	4	10	6	0.47
13	5	7	8	6	0.20
14	2	4	6	7	0.18
14	5	6	7	4	0.26

Operator	Cell	Station	Preference	Skill	Final Assignment
14	5	7	8	6	0.55
15	3	7	9	10	0.41
15	4	3	8	8	0.12
15	4	5	7	8	0.46
16	3	9	10	6	0.31
16	4	7	10	7	0.69
17	3	8	8	8	0.99
18	3	1	8	7	0.10
18	3	2	10	9	0.83
18	4	5	7	7	0.06
19	4	1	10	8	0.79
19	5	3	7	7	0.21
20	3	1	8	7	0.54
20	3	3	10	7	0.25
20	5	6	6	6	0.21
21	1	5	9	6	0.50
21	3	6	8	7	0.49
22	5	5	7	6	0.82
22	5	7	5	8	0.18
23	1	4	10	10	0.83
23	4	5	7	7	0.17

Figure 20 shows what the skill-satisfaction of each operator in the system would be if the operators would to receive the cross-training recommendations. It is appreciated that the total system skill-satisfaction has increased in comparison with not having the cross-training recommendations performed. The figure shows the skill-satisfaction for both the second stage and the third stage of the model, which are respectively the maximization of the minimum skill-satisfaction, and the maximization of the total system skill-satisfaction.

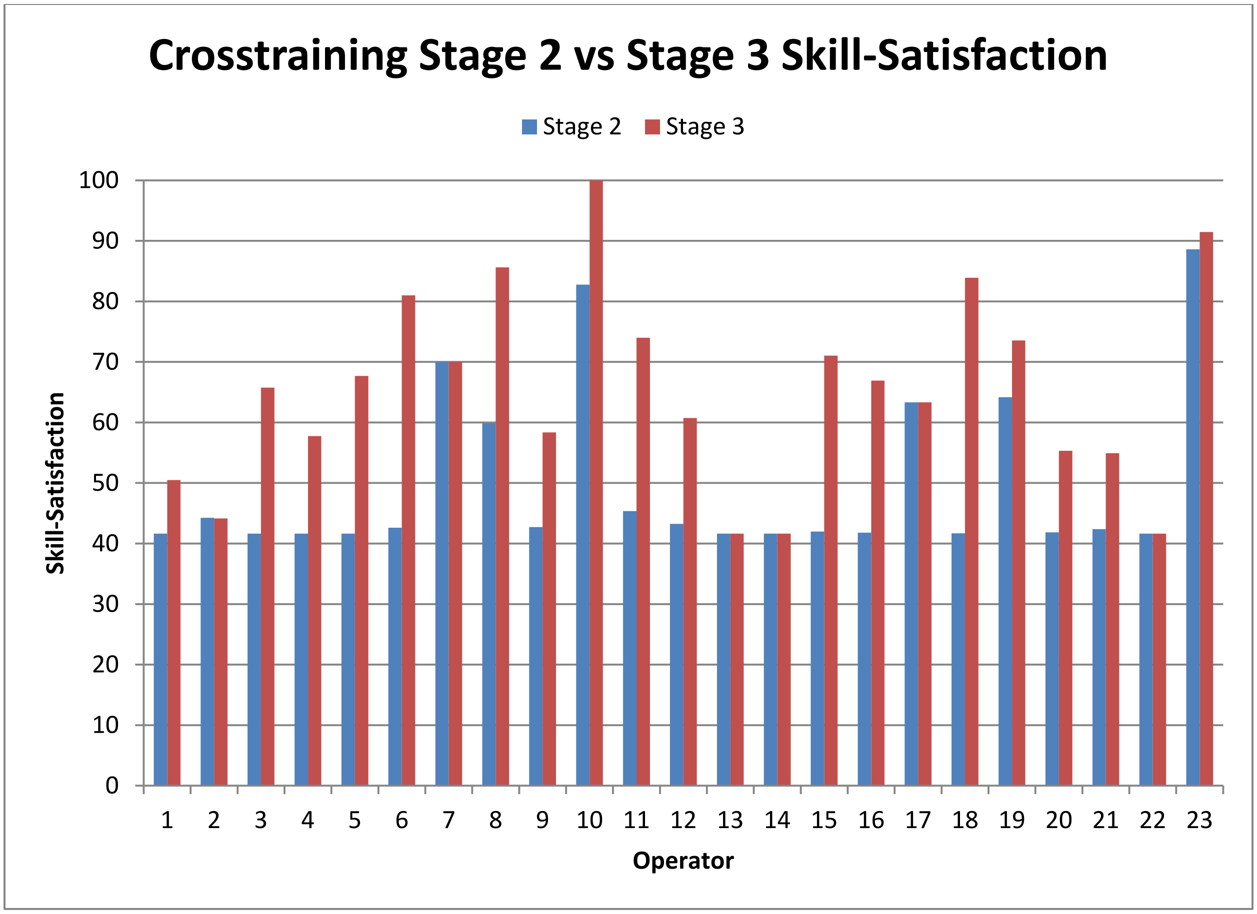

Figure 20. Skill-Satisfaction by operator for stages 2 and 3 after cross-training of the intercell policy

Table 13 presents a comparison of the minimum skill-satisfaction of the system, the total system skill-satisfaction, and whether or not the system has cross-training recommendations.

Table 13. Cross-training and no cross-training objective functions performance for intercell policy

	No Cross-training		Cross-training	
Stage	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction
2	21.88	38.21	41.65	49.92
3	21.88	56.24	41.65	65.26

4.3.2 Intracell Assignment Policy Results

The intracell case is composed of 26 operators, and as in the intercell case, the results without the cross-training recommendations will be presented first, followed by the results achieved after implementing cross-training recommendations.

4.3.2.1 Intracell No Cross-training Results

For the intracell assignment policy there are 26 operators, instead of the 23 used for the intercell assignment policy, due to the fact that in an intracell movement policy the amount of operators needed is determined by the load on each cell and not by the combined load in the whole system. With regards to the bottleneck stations we can see that all the assignments were the same as in the intercell case. Table 14 shows the bottleneck assignments for each cell, along with its corresponding associated skill and preference.

Table 14. Bottleneck stations assignments with skills and preferences for the intracell policy

Cell	Bottleneck Station	Operator	Assignment	Skill	Preference
1	4	20	0.83	7	10
2	2	15	0.81	8	9
3	8	17	0.99	8	8
4	1	19	0.79	8	10
5	1	7	1.00	7	10

By looking at Figure 21 we can see for cell 1 that even when there are now three additional operators, the operator 20 selected still possesses the highest skill among all the operators.

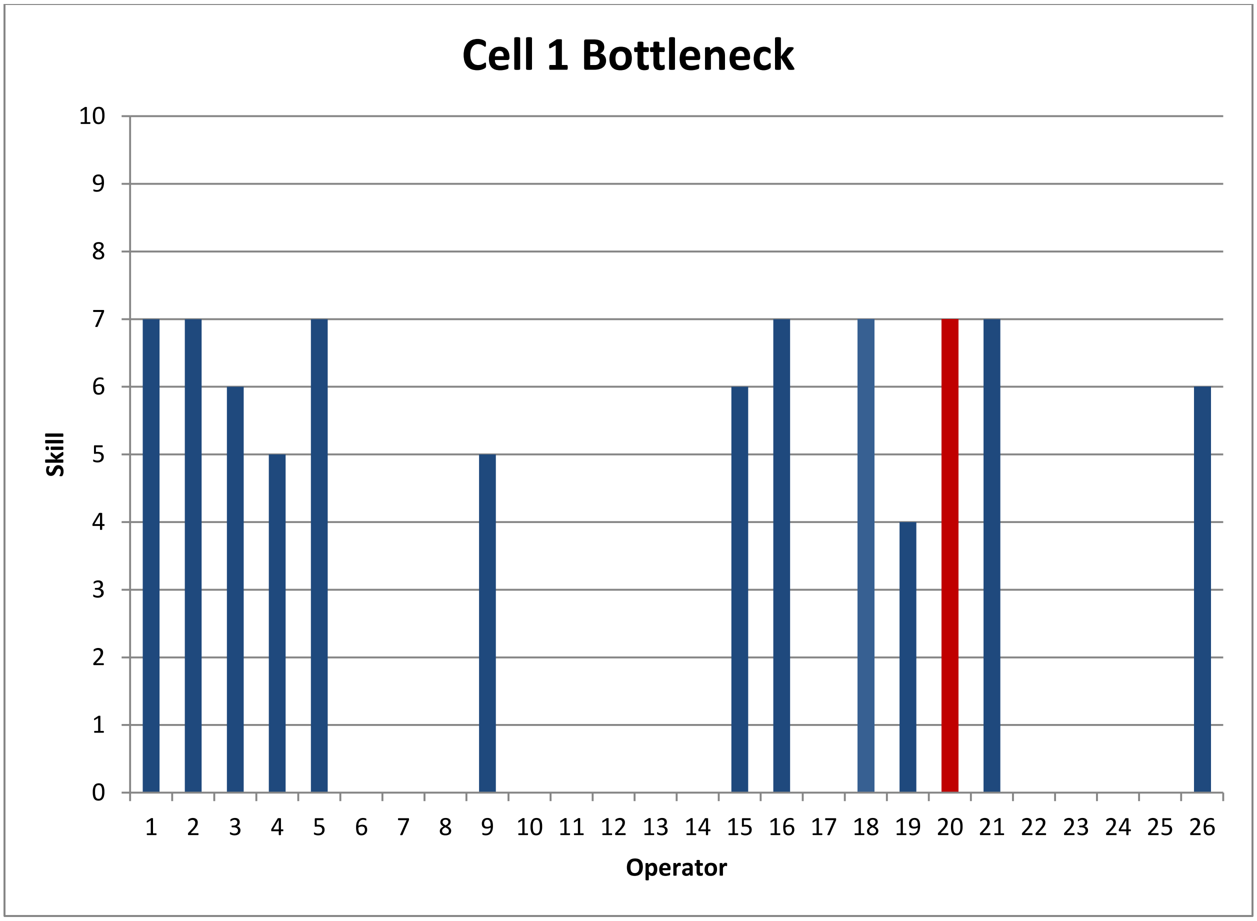

Figure 21. Bottleneck assignments for cell 1 of the intracell policy

Once the bottleneck assignments were made the model proceeded to make the rest of the assignments by leaving the bottleneck assignments fixed, and with the objective of maximizing the minimum skill-satisfaction between all the operators in the system. Lastly, with the minimum skill-satisfaction maximized, the model proceeded to input that minimum as a constraint into the next stage, which is maximizing the total system skill-satisfaction. Figure 22 shows how the total system skill-satisfaction increases from the second stage to the third stage while leaving the minimum satisfaction equal.

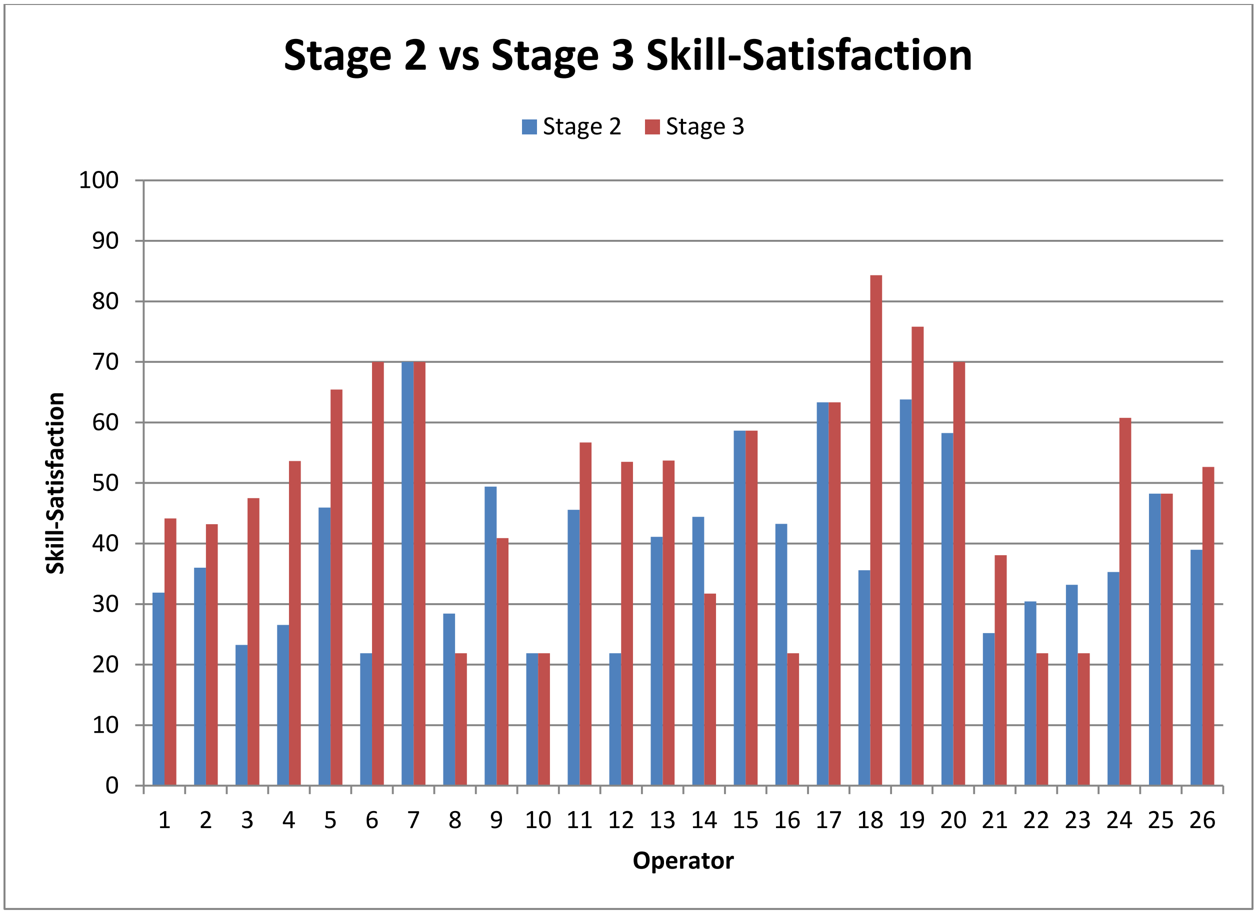

Figure 22. Skill-satisfaction by operator for stages 2 and 3 of the intracell policy

The change in the total system skill-satisfaction from one stage to another can be clearly appreciated. Table 15 summarizes the numbers for these two stages, seeing how by applying the third stage, the total system skill-satisfaction was increased from 40.09 to 49.68 while leaving the minimum skill-satisfaction at the maximum possible value.

Table 15. Stages performance for intracell policy

Stage	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction
2	21.88	40.09
3	21.88	49.68

Lastly, Table 16 shows the resulting none zero final assignments for the model. Along with the assignments, the skill and preference on the assignment cell station are presented. It is very simple to notice that both skill and preference values for the assignments are always maintained relatively high, mostly at values of five or higher.

Table 16. Preferences, skills, and final assignments for intracell policy

Operator	Cell	Station	Preference	Skill	Final Assignment
1	1	2	8	7	0.06
1	1	6	7	8	0.73
2	5	2	7	6	0.80
2	5	6	6	8	0.20
3	5	3	8	5	0.63
3	5	5	10	6	0.38
4	4	3	7	7	0.39
4	4	6	8	7	0.15
4	4	8	10	7	0.38
5	3	3	9	7	0.73
5	3	5	9	8	0.27
6	4	2	10	7	0.63
6	4	7	10	7	0.37
7	5	1	10	7	1.00
8	4	5	4	7	0.37
8	4	7	5	7	0.33
9	1	3	10	7	0.58
10	2	1	1	7	0.35
10	2	3	5	6	0.65
11	3	4	10	7	0.39
11	3	7	5	7	0.32
11	3	9	9	7	0.29
12	4	4	6	7	0.48
12	4	6	8	8	0.52
13	5	4	10	6	0.47
13	5	7	8	6	0.52
14	5	6	7	4	0.57
14	5	7	8	6	0.33
15	2	2	9	8	0.81
16	3	9	10	6	0.36
17	3	8	8	8	0.99
18	3	1	8	7	0.17

Operator	Cell	Station	Preference	Skill	Final Assignment
18	3	2	10	9	0.83
19	4	1	10	8	0.79
19	4	3	10	6	0.21
20	1	1	10	7	0.17
20	1	4	10	7	0.83
21	3	5	10	6	0.25
21	3	6	8	7	0.24
21	3	7	7	5	0.28
22	5	5	7	6	0.45
22	5	7	5	8	0.07
23	4	4	4	8	0.20
23	4	5	7	7	0.32
24	3	1	10	6	0.74
24	3	6	9	7	0.26
25	2	1	10	5	0.39
25	2	4	8	6	0.60
26	1	1	5	8	0.10
26	1	2	10	7	0.29
26	1	5	10	5	0.57

4.3.2.2 Intracell with Cross-training Results

This section presents the bottleneck assignments, final assignments, and resulting skill training, along with the new minimum and total skill-satisfaction of the intracell system with the cross-training recommendations incorporated. Out of the 26 operators in the intracell system, the model provided skill training to 52 different operator cell and station combinations across the system, and 5 operators received training in new cells. The cross-training budget was used completely, having a cross-training cost of $1,000. Table 17 presents the operators that received training in new cells.

Table 17. Operators receiving new cell training for the intracell policy

Operator	Cell
10	3
15	3
17	1
23	1
25	4

It can be seen that there was new cell training in cells 1, 3, and 4. Table 18 presents the operator and cell station where skill training was given, along with the preference, previous skill, and resulting new skill level of the operator at the given cell station.

Table 18. Cross-training data along with preferences for the intracell policy

Operator	Cell	Station	Preference	Skill	Skill Training	New Skill
1	3	4	10	6	4	10
3	1	4	9	6	3	9
3	2	3	10	8	2	10
4	1	4	9	5	5	10
6	2	3	10	7	3	10
8	4	4	10	2	8	10
10	2	2	10	5	5	10
10	3	1	2	0	1	1
10	3	2	9	0	1	1
10	3	3	9	0	1	1
10	3	4	2	0	1	1
10	3	5	3	0	1	1
10	3	6	5	0	1	1
10	3	7	10	0	10	10
10	3	8	1	0	1	1
10	3	9	2	0	1	1
11	3	4	10	7	3	10
15	1	4	10	6	4	10
15	2	2	9	8	2	10
15	3	1	3	0	1	1
15	3	2	9	0	1	1
15	3	3	5	0	1	1
15	3	4	7	0	1	1
15	3	5	5	0	1	1

Operator	Cell	Station	Preference	Skill	Skill Training	New Skill
15	3	6	3	0	1	1
15	3	7	9	0	10	10
15	3	8	6	0	1	1
15	3	9	2	0	1	1
15	4	4	9	6	4	10
15	4	8	10	6	1	7
17	1	1	9	0	1	1
17	1	2	7	0	1	1
17	1	3	2	0	1	1
17	1	4	10	0	10	10
17	1	5	7	0	1	1
17	1	6	6	0	1	1
18	4	8	10	5	1	6
20	1	4	10	7	3	10
23	1	1	5	0	1	1
23	1	2	6	0	1	1
23	1	3	10	0	1	1
23	1	4	10	0	10	10
23	1	5	8	0	1	1
23	1	6	9	0	1	1
25	4	1	4	0	1	1
25	4	2	9	0	1	1
25	4	3	6	0	1	1
25	4	4	10	0	10	10
25	4	5	9	0	1	1
25	4	6	2	0	1	1
25	4	7	4	0	1	1
25	4	8	7	0	1	1

With regards to Table 18, it can be seen that all the skill training was performed in cell stations were the operators had high preferences. It’s important to mention that the case when the operators had a skill of zero, and a skill training of one in a station, means that the operators only received the new cell training which increased their skill level to the minimum possible of one. Nonetheless, when an operator receives new cell training, they receive additional skill training in order to increase the skill level to the minimum required in the station in at least in one of the stations of the cell. This also explains why there are operators that received a minimum skill training of one, in a station with a low preference. Revising in more detail those cases it can be appreciated that the operators received a training conducting to the minimum skill required in the station where the operator had the highest preference.

Table 19 shows the new resulting bottleneck assignments after the cross-training recommendations. It can be seen that the operator assignment changed only for cells 1 and 2. Originally, the bottleneck station in cell 1 was assigned the operator 20, and after cross-training recommendations, operator 23 was assigned. In cell 2 the operator 15 was originally assigned and changed to operator 10 after cross-training recommendations. These two changes in the assignments are due to the fact that these two operators now possess the maximum skill of ten due to the cross-training performed.

Table 19. Cross-training bottleneck assignments for the intracell policy

Cell	Bottleneck Station	Operator	Assignment	Skill	Preference
1	4	23	0.83	10	10
2	2	10	0.81	10	10
3	8	17	0.99	8	8
4	1	19	0.79	8	10
5	1	7	1.00	7	10

Table 20 shows the final assignments after implementing the cross-training recommendations provided by the model.

Table 20. Cross-training final assignments for the intracell policy

Operator	Cell	Station	Preference	Skill	A
1	5	3	8	6	0.63
1	5	5	6	7	0.11
1	5	6	4	6	0.26
2	5	2	7	6	0.80

Operator	Cell	Station	Preference	Skill	A
2	5	6	6	8	0.15
3	1	2	10	7	0.19
3	1	6	8	6	0.66
4	4	3	7	7	0.59
4	4	8	10	7	0.17
5	3	3	9	7	0.73
5	3	5	9	8	0.27
6	2	3	10	10	0.65
6	2	4	9	6	0.35
7	5	1	10	7	1.00
8	4	4	10	10	0.67
8	4	5	4	7	0.32
9	1	3	10	7	0.58
9	1	6	6	7	0.06
10	2	2	10	10	0.81
11	3	4	10	10	0.39
11	3	9	9	7	0.59
12	4	6	8	8	0.67
13	5	4	10	6	0.16
13	5	7	8	6	0.65
14	4	2	7	6	0.63
14	4	5	4	8	0.16
14	4	8	9	5	0.20
15	3	7	9	10	0.60
16	4	7	10	7	0.69
17	3	8	8	8	0.99
18	3	2	10	9	0.83
19	4	1	10	8	0.79
19	4	5	5	7	0.21
20	5	4	10	9	0.31
20	5	6	6	6	0.36
21	3	5	10	6	0.25
21	3	6	8	7	0.41
21	3	9	10	5	0.06
22	5	5	7	6	0.71
22	5	7	5	8	0.27
23	1	4	10	10	0.83
24	3	1	10	6	0.91
24	3	6	9	7	0.09
25	2	1	10	5	0.74
25	2	4	8	6	0.25

Operator	Cell	Station	Preference	Skill	A
26	1	1	5	8	0.27
26	1	2	10	7	0.17
26	1	5	10	5	0.57

Figure 23 shows what the skill-satisfaction of each operator in the system would be if the operators would to receive the cross-training recommendations. It is shown that the total system skill-satisfaction has increased in comparison with not having the cross-training recommendations performed. The figure shows the skill-satisfaction for both the second stage and the third stage of the model, which are respectively the maximization of the minimum skill-satisfaction, and the maximization of the total system skill-satisfaction.

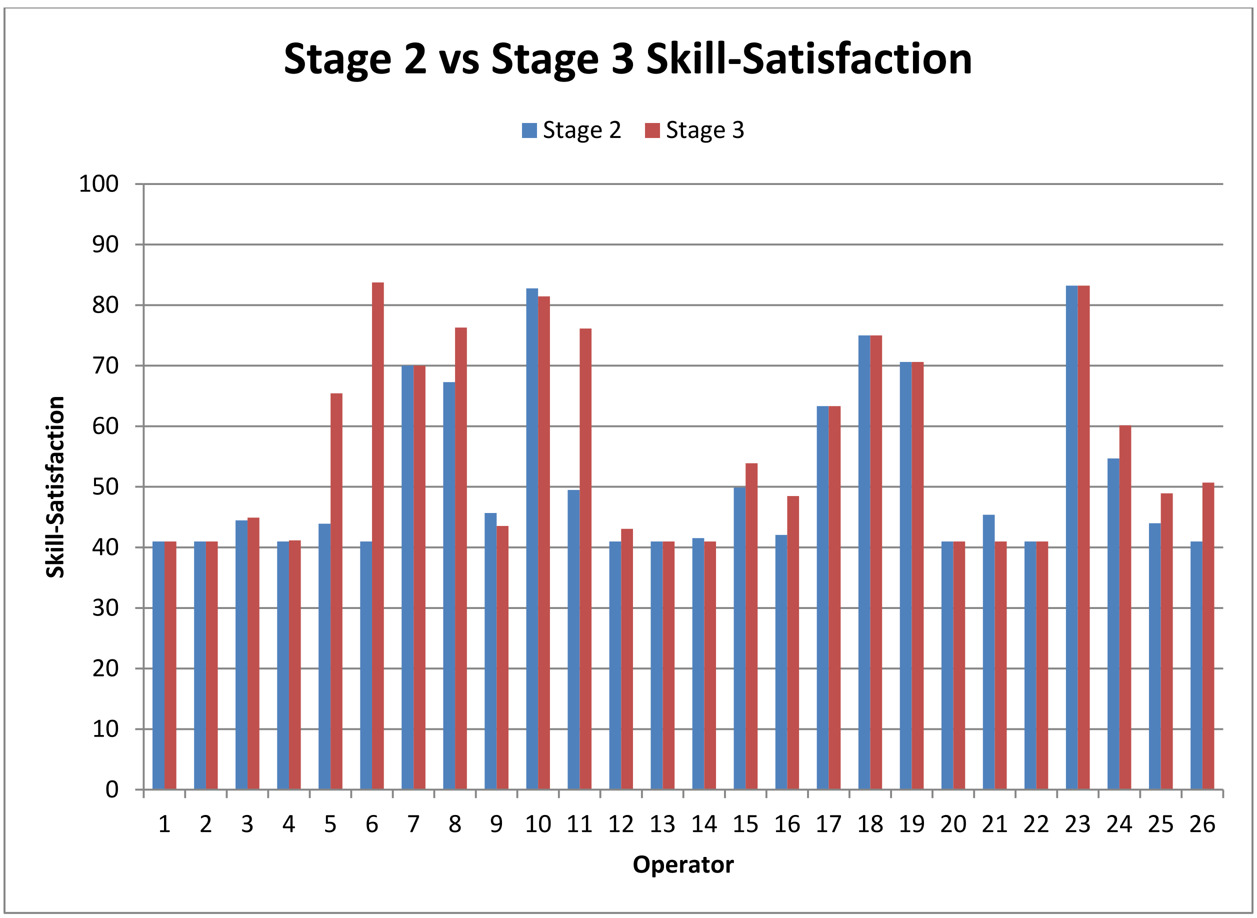

Figure 23. Skill-satisfaction by operator for stages 2 and 3 after cross-training of the intracell policy

Table 21 presents a comparison of the minimum skill-satisfaction of the system, the total system skill-satisfaction, and whether or not the system has cross-training recommendations.

Table 21. Cross-training and no cross-training objective functions performance for intracell policy

	No Cross-training		Cross-training	
Stage	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction	Minimum Op Skill-Satisfaction	Total System Skill-Satisfaction
2	21.88	40.09	40.98	51.62
3	21.88	49.68	40.98	56.42

4.3.3 Results Comparison

With the purpose of comparing the results of the intracell and intercell scenarios along with cross-training recommendations, Table 22 shows the most relevant numbers for all the cases.

Table 22. Comparison of intracell, intercell and cross-training scenarios for each model stage

	No-Cross-training				Cross-training			
Assignment Policy	Intercell		Intracell		Intercell		Intracell	
Model Stage	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**
Minimum skill-satisfaction	21.88	21.88	21.88	21.88	41.65	41.65	40.98	40.98
Total system satisfaction	38.21	56.24	40.09	49.68	49.92	65.26	51.62	56.42

Table 22 shows that the highest minimum skill-satisfaction obtained was 41.65 achieved by completing cross-training in the intercell scenario. Additionally, it is seen that as the model intended, the total system skill-satisfaction was considerably increased while leaving the minimum skill-satisfaction between all operators at the maximum possible value. The following 4 Figures show the changes in the skill-satisfaction due to cross-training, for the stages of the intercell and intracell cases. First, the changes for the two stages of the intercell assignment policy are presented.

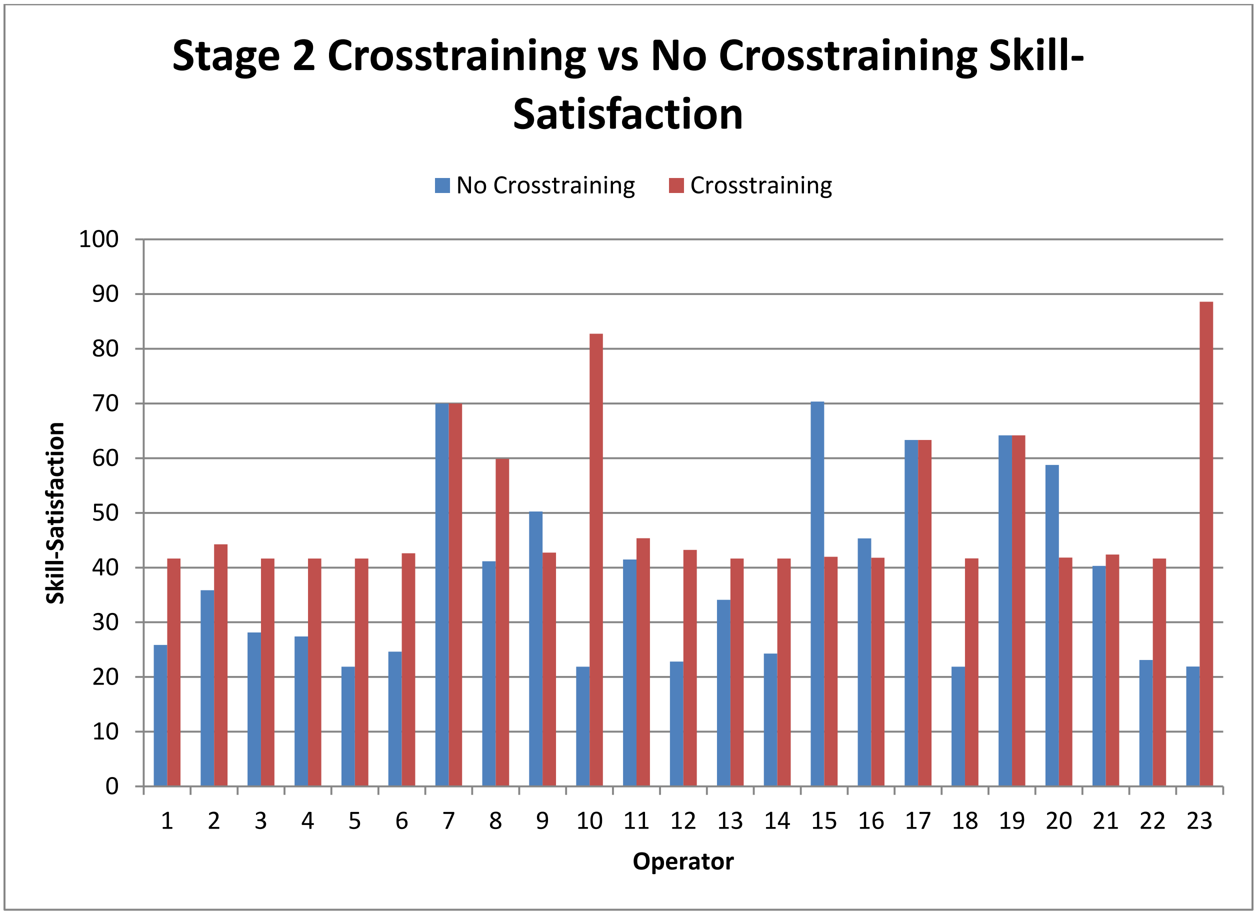

Figure 24. Skill-satisfaction for the second stage of the intercell policy

Figure 24 shows the skill-satisfaction for the second stage objective function in the intercell assignment policy. The figure shows the changes in skill-satisfaction due to the presence of cross-training recommendations. In this figure the main detail to observe is the increase in the minimum skill-satisfaction in the system, which increased from 21.88 to 41.65. The total skill-satisfaction also increased with the cross-training recommendations, even when it does not represent any relationship with the cross-training, since the purpose of this stage is only to maximize the system minimum skill-satisfaction.

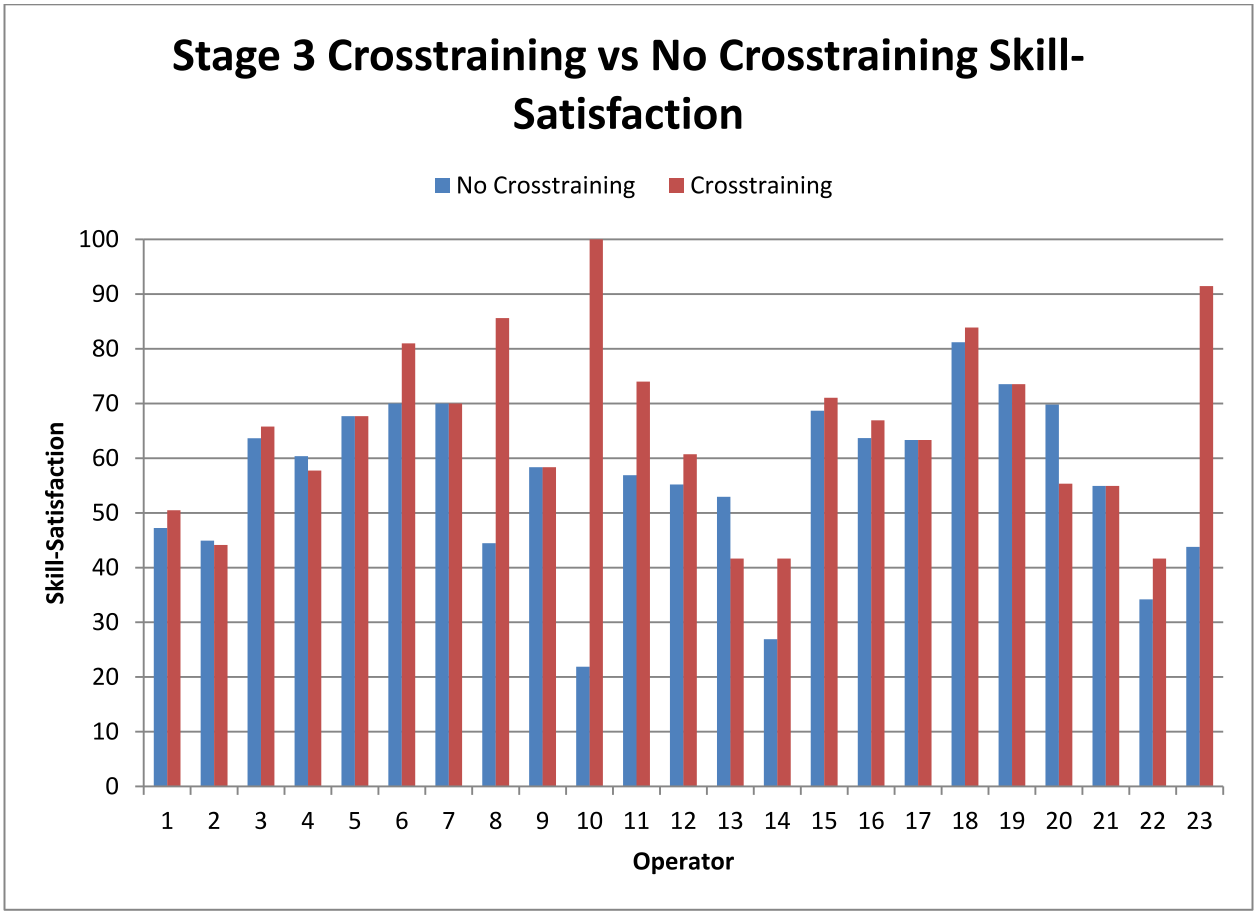

Figure 25. Skill-satisfaction for the third stage of the intercell policy

Figure 25 shows the skill-satisfaction for the third stage objective function in the intercell assignment policy. The figure presents the changes in skill-satisfaction due to the presence of cross-training recommendations. In this figure the main detail to observe is the increase in the total system skill-satisfaction, which increased from 56.24 to 65.26. The minimum system skill-satisfaction remained the same as in the second stage with cross-training recommendations since the purpose of the third stage is to increase the total system skill-satisfaction while leaving the minimum skill-satisfaction at its maximum possible value.

The next two graphs present the changes for the two stages of the intracell assignment policy. The first one represents the changes due to cross-training in the second stage of the model, while the second one shows the changes due to cross-training in the third stage of the model.

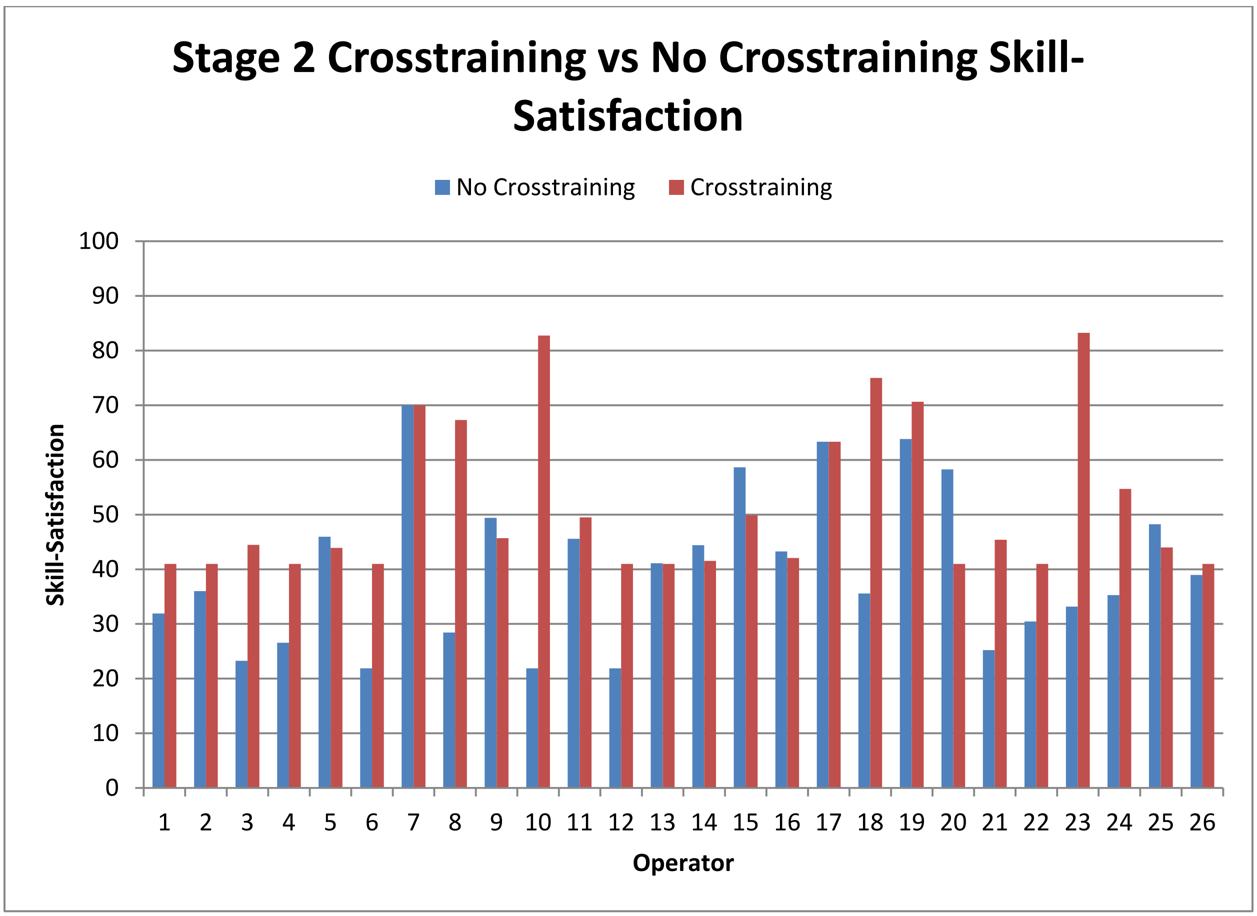

Figure 26. Skill-satisfaction for the second stage of the intracell policy

Figure 26 shows the skill-satisfaction for the second stage objective function in the intracell assignment policy. The figure shows the changes in skill-satisfaction due to the presence of cross-training recommendations. In this figure the main detail to observe is the increase in the minimum skill-satisfaction in the system, which increased from 21.88 to 40.98. Here we can see that even when the system minimum skill-satisfaction increased, the increase was less than in the intercell case due to the fact that in the intracell assignment policy the operators are confined to only one cell, giving less flexibility to the model for maximizing the objective function.

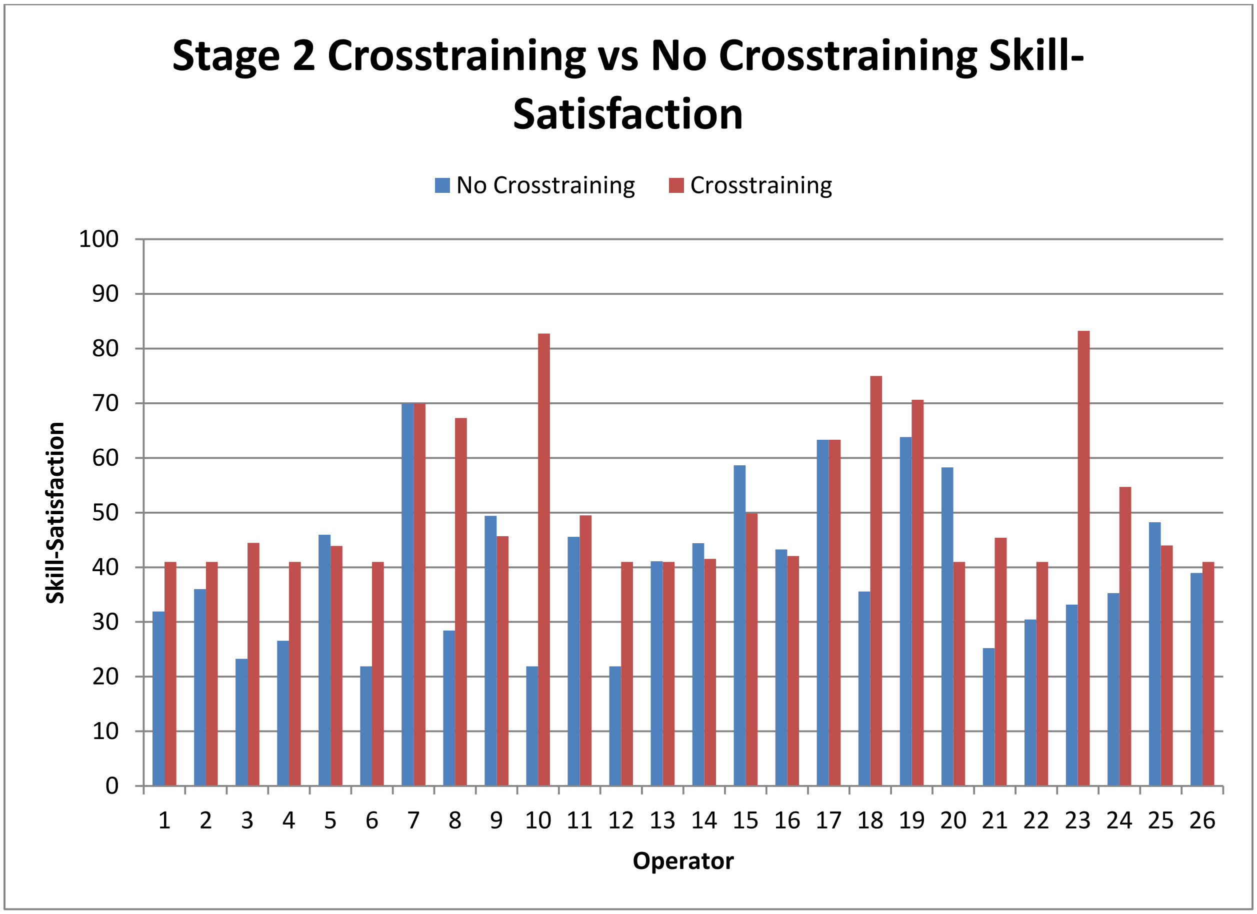

Figure 27. Skill-satisfaction for the third stage of the intracell policy

Figure 27 shows the skill-satisfaction for the third stage objective function in the intracell assignment policy. In this figure the main detail to observe is the increase in the total system skill-satisfaction, which increased from 49.68 to 56.42. Again we see here that even when the total system skill-satisfaction increased, the increase was lower in comparison to the intercell policy scenario, this due to the less flexibility provided by the intracell assignment policy.

For this case study we can conclude that that the best assignment policy would be an intercell policy, because it brings the highest total system skill-satisfaction. Moreover, if the cross-training recommendations are performed, the intercell scenario would still be the best assignment policy since it would still bring the highest total system skill-satisfaction. This is easily explained by the fact that with an intercell policy the model has a higher flexibility with regards to the assignments. With an intercell movement policy operators can be assigned between all the cells in the system, giving more chance to the model to find a higher total skill-satisfaction assignment arrangement even when there are less operators than in the intracell policy scenario.

Chapter 5

This chapter provides a brief conclusion of the work previously discussed, followed by different alternatives that are considered as possible future work opportunities to expand the scope of this work.

5.1 Conclusions

This work proposed a linear programming model that takes into account the operators' skills and preferences in order to maximize the resulting skill-satisfaction of their assignments. It resolves the important issue of lack of consideration of operators' preferences as a critical criteria in the labor assignment decision making process. The research results demonstrated that labor assignments considering both skills and preferences not only can satisfy system's demand but also can increase operators' satisfaction regarding task assignments.

The methodology used allowed the consideration of non-technical characteristics such as operator's preference while ensuring that the technical requirements of the workstations were satisfied. The skills parameter was responsible for covering the technical characteristics of the model, with more attention at the bottleneck stations model stage. Since it is clear that the technical aspects of the operators are important for the assignments, differently from the rest of the model the bottleneck stations cover mainly the skills with a particular consideration to operator preferences in order to maximize the system skilled-satisfaction.

The skill-satisfaction metric created provides a way of measuring a skill – preference combination, which becomes a skilled satisfaction measure due to the time-assignment weight used. This metric provides a means for measuring a non – quantitative characteristic in order to assess how happy or satisfied are the operators, without never lacking consideration of their technical characteristics.

Through the case study presented it was shown that indeed the model developed enhances the minimum skill-satisfaction and total system skill-satisfaction of operators. It was shown how the minimum technical requirements of the system were always fulfilled while maximizing to the greatest extent the preferences of the operators. In conclusion we have shown that this model could be an acceptable and useful tool in the management decision making process of any company.

5.2 Future Work

Even though the scope of this work was completely covered, there are still topics that can be considered as possible future research to extend the scope of this work. The following topics are considered as potential future work.

- Provide scheduling results in combination with labor assignments.

This work provides the assignment of operators and not the scheduling or order of the different tasks that these operators will perform. When an operator is partially assigned to several stations, it is understood that the operator will be working a certain percent of the time in a station and the rest in another station(s). The problem of scheduling, i.e. the order in which the operators will work in those stations, was not addressed here, but it is considered as a strong topic for future research.

- Modify the model for a Semi-Automatic Manufacturing Environment

One of the limitations of this work is that the model applies to a system where cells perform only manual labor. This limitation is a great opportunity for expanding the scope of this work. The current model could be modified to include the consideration of semiautomatic machines, where machines have to be loaded and unloaded, with an associated machine cycle time. The inclusion of this modification would open this model to a wider range of manufacturing plants.

- Modify the model to include Overtime Assignments

The model's main decision variable is the amount of time an operator will be assigned to a specific task. One of the constraints the model possesses is that no operator can be assigned more than one complete work shift. A possible opportunity to modify the model is to allow it to assign operators in more than one shift, what would mean assigning operators overtime. This modification could be achieved by associating an overtime cost with the overtime time assignment, and adding a constraint on how much overtime will be allowed based on a previously imposed budget.

- Modify the model to focus on the performance of cells and not the whole system.

This model offers the option to select between an intercell and an intracell movement assignment policy. In an intracell movement assignment policy, cells are treated as independent systems. However the objective functions in the model seek to improve the performance of the system, instead of the performance of each cell individually. The model could be modified so that when the user selects an intracell movement assignment policy, the objective functions seek to maximize the operators' minimum skill-satisfaction by cell, and the total system skill-satisfaction by cell.

- Modify cross-training model in order to train on a need basis.

Currently, the cross-training model provides training recommendations based on the preferences of the operators. This means that the higher the preference of an operator is in a specific station, the more probable it will be for the operator to receive training in that station. A possible modification that could be made to the cross-training model is to train operators on a need basis, which means that the model will provide training recommendations based on where the skill-satisfaction metric is the lowest. Although currently the skill-satisfaction metric is calculated for each operator, it could be easily calculated for each cell in order to accommodate this modification.

- Include the displacement time among cells/stations in the model formulation.

One of the main assumptions of the model developed is that cells and stations are located at near distances from each other; i.e. displacement times are negligible. This assumption is introduced because the model created offers the option to assign operators to more than one cell, and more than one station, and we need to ensure that the distance between cells and stations is close enough for an operator to travel from one cell/station to another in a considerably small amount of time. Instead of assuming negligible distances the model could be modified to include a displacement time to the assignment total time at the cell stations. In other words each operator should be able to work one complete work shift, which is composed of the displacement time plus the time the operator works at the different cell stations. This displacement time should vary depending on the operator cell station assignment because all cells and stations are not located at the same distances. For example if an operator is assigned to two different stations, then the model should add the displacement time it takes to travel from one station to another once. On the other hand if the model assigns an operator to three different stations, then the model should add two displacement times.

- Maximize the system cells production prior to maximizing the system skill – satisfaction assignments.

The developed model seeks to create an assignment that maximizes the system skill – satisfaction, and this assignment is based on a capacity analysis that is calculated *a priori* and estimates how many operators are needed to comply with the required cell demands. The model will assign the calculated amount of operators to satisfy the required system cell demands. The problem with this is that most of the time the amount of operators calculated is actually able to produce more than the required demand. An extra model stage that should be executed before all the stages could be created in order to maximize the system cell demand according to the amount of operators needed in the system. In this way we would avoid having excess of operators' idle time.

- Modify the model to include additional non-technical characteristics.

Additional non-technical characteristics such as operator preferences for a product line, the operator preferences for working with a coworker or group of coworkers, and the absenteeism record of the operators are some characteristics that could be included in the model.

- Create a training reward program to use in combination with the assignment model.

The cross-training model created facilitates the establishment of an operators' overall training metric. By creating a metric that represents an operators overall system training, a reward program could be developed in order to reward operators based on their overall training or knowledge in the system. The purpose of the reward program would be to offer greater rewards to operators with the highest knowledge or training in the system.

- Perform a correlation analysis on the skill and preference parameters with the objective function.

The skill and preference parameters are key factors on the performance of the model objective functions. The values that these two parameters possess can decide how high or low the objective function will get, or even if our model will be feasible or not. For this reason it is recommended an analysis of correlation between these two parameters and the objective function. Different case scenarios with different levels for these two parameters could be created with the purpose of evaluation of the objective function performance. For example the skill and preference parameters could be set to have the following characteristics: High skills Vs Low Preferences, and Low skills Vs High preferences.

Bibliography

1. Bidanda, B., Ariyawongrat, P., Needy, K. L., Norman, B. A., & Tharmmaphornphilas, W. (2005). Human related issues in manufacturing cell design, implementation, and operation: a review and survey. Computers & Industrial Engineering , 48, 507–523.

2. Bokhorst, J. A., & Slomp, J. (2007). Design and operation of a cross-trained workforce. In P. S. University, & D. Nembhard (Ed.), Workforce Cross Training (pp. 3-63). CRC Press.

3. Brusco, M. J. (2008). An exact algorithm for a workforce allocation problem with application to an analysis of cross-training policies. IIE Transactions , 40, 495–508.

4. Cesaní, V. I., & Steudel, H. J. (2005). A study of labor assignment flexibility in cellular manufacturing systems. Computers & Industrial Engineering , 48, 571–591.

5. DePuy, G. W., Grieshaber, D., & Hardin, C. T. (2009). Skills Management Assignment Problem Heuristics. *Proceedings of the 2009 Industrial Engineering Research Conference*, (pp. 1994-1999). Orlando, Florida.

6. Egilmez, G., & Suer, G. A. (2011). Stochastic Manpower Allocation and Cell Loading in Cellular Manufacturing Systems. Proceedings of the 41st International Conference on Computers & Industrial Engineering, (pp. 193-198). Loas Angeles, CA USA.

7. Fitzpatrick, E. L., & Askin, R. G. (2005). Forming effective worker teams with multi-functional skill requirements. Computers & Industrial Engineering , 48, 593–608.

8. Hoo, Y. C., & Moodie, C. L. (1996). Solving Cell Formation Problems in a Manufacturing Enviroment with Flexible Processing and Routing Capabilities. International Journal of Production Research , 34, 2901-2923.

9. Hyer, N. L., & Brown, K. A. (2003). Work Cells with Staying Power: LESSONS FOR PROCESS COMPLETE OPERATIONS. California Management Review , 46, 27-52.

10. Inman, R. R., Jordan, W. C., & Blumenfeld, D. E. (2004). Chained cross-training of assembly line workers. International Journal of Operations Research , 42, 1899–1910.

11. Kattan, I. A. (1997). Design and scheduling of hybridmulti-cell flexible manufacturing systems. International Journal of Production Research , 35, 1239-1257.

12. Kher, H. V., & Fry, T. D. (2001). Labour flexibility and assignment policies in a job shop having incommensurable objectives. International Journal of Production Research , 39, 2295-2311.

13. Liu, C., Yasuda, K., Yin, Y., & Tanaka, K. (2009). Uncertain association rule mining algorithm for the cell formation problem in cellular manufacturing systems. International Journal of Production Research , 47, 667–685.

14. McDonald, T., Ellis, K. P., Aken, E. M., & Koelling, C. P. (2009). Development and application of a worker assignment model to evaluate a lean manufacturing cell. International Journal of Production Research , 47, 2427-2447.

15. Nembhard, D. A., & Norman, B. A. (2007). Cross Training in production systems with human learning and forgetting. In P. S. University, & D. Nembhard (Ed.), Workforce Cross Training (pp. 111-129). CRC Press.

16. Niebel, B., & Freivalds, A. (2003). Methods, Standards and Work Design (Vol. 11). New York: McGraw-Hill.

17. Pastor, R., & Corominas, A. (2007). Job assignment. In P. S. Univerisity, & D. Nembhard (Ed.), Workforce Cross training (pp. 65-85). CRC Press.

18. Robles-Leon, W.. (2013). Understanding operator preferences towards manufacturing cell task assignments. Mayaguez, PR: Univerisity of Puerto Rico at Mayaguez.

19. Slomp, J., Bokhorst, J. A., & Molleman, E. (2005). Cross-training in a cellular manufacturing environment. Computers & Industrial Engineering , 48, 609–624.

20. Suer, G. A., & Alhawari, O. (2013). Operator Assignment Decisions in a Highly Dynamic Cellular Environment. In I. Management Association, USA (Ed). Industrial Engineering: Concepts, Methodologies, Tools, and Applications (pp. 1135-1152). Hershey, PA: Engineering Science Reference. doi:10.4018/978-1-4666-1945-6.ch061

21. Süer, G. A., & Dagli, C. (2005). Intra-cell manpower transfers and cell loading in labor-intensive manufacturing cells. Computers & Industrial Engineering , 48, 643–655.

22. Süer, G. A., & Sanchez-Bera, I. (1998). Multi-Period Cell Loading and Cell Size Determination. Computers & Industrial Engineering , 35, 85-88.

23. Süer, G. A., Cosner, J., & Patten, A. (2009). Models for cell loading and product sequencing in labor intensive cells. Computers & Industrial Engineering , 56, 97–105.

24. Suresh, N. C., & Slomp, J. (2001). A multi-objective procedure for labour assignments and grouping in capacitated cell formation problems. International Journal of Production Research , 39, 4103-4131.

25. Yang, K. K. (2007). A comparison of cross-training policies in different job shops. International Journal of Production Research , 45, 1279–1295.

26. Yue, H., Slomp, J., Molleman, E., & Zee, D. J. (2008). Worker flexibility in a parallel dual resource constrained job shop. International Journal of Production Research , 46, 451–467.

Appendices

A. Glossary

The Glossary introduces definitions to concepts utilized throughout this document.

- **Skills -** The skills parameter refers to how skilled or productive an operator is in a specific area. The skills parameter is a measure that every company has to define in its way. For example, for one company the skill of an operator could be defined by the amount of time the operator has been working the process, or the knowledge the operator has in the process. On the other hand, another company could define it as the productivity numbers related to the operator in the process. The end goal is that the metric selected by the company has to be converted to a zero through ten scale, where ten is the best skill possible, and zero means that the operator has no training in the cell.
- **Preference** – Operator's preference in the context of this research is a parameter that tells how interested an operator is in a specific area. These preferences could be achieved by distributing surveys that focus on the operator's preferences in the system, and finally converting these surveys to a one through ten scale, where ten is the best preference possible.
- **Satisfaction –** Satisfaction is a measure of fulfillment for an operator due to their assignment to a specific task. It takes into consideration the preference of the operator in a station along with the amount of time the operator was assigned to the station. It is based on the fact that while more time you spend on a task that you like, the more satisfied you'll be.
- **Skill-Satisfaction –** Skill-Satisfaction is a measure of skill fulfillment for an operator due to their assignment to a specific task. It takes into consideration the preference and skill of the operator in a station along with the amount of time the operator was assigned to the station.

The most appropriate scenario would be to assign an operator to a station where it has the highest combination of preference and skill.

- **Shift Percent –** “Shift percent” is a percent time measure of the total time of a shift. It represents the amount of time a station has to be covered in order to comply with the imposed demand in the cell.
- **Cellular manufacturing -** It’s a workplace design model that seeks to take full advantage of the similarity between parts, through standardization and common processing.
- **Cross-training** – It’s a change in the traditional way of thinking where employees have specific individual job descriptions. The basic purpose is to have multiple individuals trained in various functions which not only makes an employee more valuable, but it also helps employees better themselves.
- **Bottleneck Station -** These are stations that limit production in comparison with the rest of the stations inside a cell. Since cells operate as independent manufacturing environments, each cell contains its bottleneck station, mostly due to higher standard times or a complex operation.
- **Intracell -** Intracell flexibility relates to operator’s transfers between machines within a cell. When operators are assigned to stations in an intracell manufacturing policy, each operator cannot be assigned to more than one cell at the same time.
- **Intercell** - Intercell labor flexibility refers to the transfer of operators between cells. When operators are assigned to stations in an intercell manufacturing policy, these can be assigned to more than one cell in a system at the same time.
- **Machine Intensive Cell -** In a machine intensive cell the number of machines is the primary parameter used on determining the output, so the impact of labor on the output is limited.

Normally, the operator's role in machine-intensive cells is limited due to the presence of automatic machines.

- **Labor Intensive Cell -** Labor-intensive manufacturing cells consist of simple machines and equipment that require continuous operator attendance and involvement, thus the number of operators and their assignment to operations have a great impact on the cell's production rate.
- **Dual Resource Constraint** – DRCs are systems where the number of machines exceeds the number of workers, and both worker and machine capacity constrain the output of the system.

B. Model Programming in Lingo

Appendix B shows the code programmed in the optimization software Lingo 11.

```
Model:

DATA:
ce =@ODBC('OptModel','Parameters','Cells');
o = @ODBC('OptModel','Parameters','Operators');
B = @ODBC('OptModel','Parameters','Budget');
M = 1000000;
w = .001;
cr = @ODBC('OptModel','Parameters','Crosstraining');
dum2 = 0;
ENDDATA

Sets:
Celdas / 1..ce /:Demand,AsigCelda,l,Sj,CTcost;
EndSets

Data:
l =@ODBC('OptModel','CellP','MinAssignment');
v=@ODBC('OptModel','Parameters','MaxCells');
h=@ODBC('OptModel','Parameters','MaxStations');
Sj = @ODBC('OptModel','Stations','Stations');
CTcost =@ODBC('OptModel','CellP','CTCost');
End Data

Calc:
ej= @Max(Celdas(j):Sj(j));
EndCalc

Sets:
Operadores / 1..o /: S,Asig,Skill,Pref,Dummy;
Estaciones / 1..ej /;
ConjuntoTotal(Operadores,Celdas,Estaciones)|&3#LE#Sj(&2):Aijk,BAijk,NAijk,Pij
k,Sijk,NSijk,Iijk,tijk,Qijk;
CeldaEstacion(Celdas,Estaciones)|&2#LE#Sj(&1):CSjk,Bjk,Djk,ACelEst,Scost ;
OperadorCelda(Operadores,Celdas):Yij,Tij,CTij,NTij,AOpCel;
OperadorEstacion(Operadores,Estaciones);
EndSets

Data:
Djk = @ODBC('OptModel','CellStaP','Demand');
Pijk =@ODBC('OptModel','OpCellStaP','Preference');
Sijk =@ODBC('OptModel','OpCellStaP','Skill');
CSjk =@ODBC('OptModel','CellStaP','Minimum Skill');
Tij =@ODBC('OptModel','OpCellP','Cell Training');
Bjk =@ODBC('OptModel','CellStaP','Bottleneck');
Scost =@ODBC('OptModel','CellStaP','Skill Cost');
end data



SubModel Botella:
Max= @sum(ConjuntoTotal(i,j,k):Aijk(i,j,k)*(Sijk(i,j,k) + w*Pijk(i,j,k))) ;
```

```
@for(CeldaEstacion(j,k):@sum(Operadores(i):Aijk(i,j,k))=Bjk(j,k)*Djk(j,k));
@for(ConjuntoTotal(i,j,k):Sijk(i,j,k)>=Iijk(i,j,k)*CSjk(j,k));
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)<=Tij(i,j));
@for(Operadores(i):@sum(CeldaEstacion(j,k):Aijk(i,j,k)) <= 1);
@for(ConjuntoTotal(i,j,k):@bin(Iijk));
@for(ConjuntoTotal(i,j,k):Iijk(i,j,k)>=Aijk(i,j,k));
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)>= l(j)*Iijk(i,j,k));
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)>=0);

@for(CeldaEstacion(j,k):@sum(Operadores(i):Iijk(i,j,k))=Bjk(j,k));
@for(Operadores(i):@sum(CeldaEstacion(j,k):Iijk(i,j,k))<=1);

@for(Operadores(i):Asig(i) = @sum(CeldaEstacion(j,k):Aijk(i,j,k)));
@for(Celdas(j):Demand(j) = @sum(Estaciones(k)|k#LE#Sj(j):Djk(j,k)));
@for(Celdas(j):AsigCelda(j) =
@sum(OperadorEstacion(i,k)|k#LE#Sj(j):Aijk(i,j,k)));
@for(OperadorCelda(i,j):AOpCel(i,j) =
@sum(Estaciones(k)|k#LE#Sj(j):Aijk(i,j,k)));
@for(CeldaEstacion(j,k):ACelEst(j,k)=@sum(Operadores(i):Aijk(i,j,k)));
@for(Operadores(i):Skill(i) =
100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Sijk(i,j,k))/(10));
@for(Operadores(i):S(i) =
100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i,j,k))/(10*10));
TSS =
100*@sum(ConjuntoTotal(i,j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i,j,k))/(10*10*o);
EndSubmodel


SubModel Smin:
Max= U ;

@for(Operadores(i):100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i
,j,k))/(10*10)>=U);
EndSubmodel



Submodel Stotal:
Max=
100*@sum(ConjuntoTotal(i,j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i,j,k))/(10*10*o)
;

@for(Operadores(i):100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i
,j,k))/(10*10)>=LB);
EndSubmodel



Submodel MinTotCons:
@for(CeldaEstacion(j,k):@sum(Operadores(i):Aijk(i,j,k))=Djk(j,k));
@for(ConjuntoTotal(i,j,k):Sijk(i,j,k)>=Iijk(i,j,k)*CSjk(j,k));
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)<=Tij(i,j));
@for(Operadores(i):@sum(CeldaEstacion(j,k):Aijk(i,j,k)) <= 1);
@for(ConjuntoTotal(i,j,k):@bin(Iijk));
@for(ConjuntoTotal(i,j,k):Iijk(i,j,k)>=Aijk(i,j,k));
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)>= l(j)*Iijk(i,j,k));
```

```
@for(ConjuntoTotal(i,j,k):Aijk(i,j,k)>=0);

@for(Operadores(i):@sum(CeldaEstacion(j,k):Iijk(i,j,k))<=h);

@for(OperadorCelda(i,j):@bin(Yij));
@for(Operadores(i):@sum(Celdas(j):Yij(i,j))<=v);
@for(OperadorCelda(i,j):@sum(Estaciones(k)|k#LE#Sj(j):Aijk(i,j,k))<=Yij(i,j))
;

@for(ConjuntoTotal(i,j,k):Aijk(i,j,k) = BAijk(i,j,k)+NAijk(i,j,k));

@for(Operadores(i):S(i) =
100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i,j,k))/(10*10));
@for(Operadores(i):Skill(i) =
100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Sijk(i,j,k))/(10));
@for(Operadores(i):Pref(i) =
100*@sum(CeldaEstacion(j,k):Aijk(i,j,k)*Pijk(i,j,k))/(10));
@for(Operadores(i):Asig(i) = @sum(CeldaEstacion(j,k):Aijk(i,j,k)));
@for(Celdas(j):Demand(j) = @sum(Estaciones(k)|k#LE#Sj(j):Djk(j,k)));
@for(Celdas(j):AsigCelda(j) =
@sum(OperadorEstacion(i,k)|k#LE#Sj(j):Aijk(i,j,k)));
@for(OperadorCelda(i,j):AOpCel(i,j) =
@sum(Estaciones(k)|k#LE#Sj(j):Aijk(i,j,k)));
@for(CeldaEstacion(j,k):ACelEst(j,k)=@sum(Operadores(i):Aijk(i,j,k)));
TSS =
100*@sum(ConjuntoTotal(i,j,k):Aijk(i,j,k)*Pijk(i,j,k)*Sijk(i,j,k))/(10*10*o);
EndSubmodel



SubModel CrossTraining:
Max= @sum(ConjuntoTotal(i,j,k):Pijk(i,j,k)*NSijk(i,j,k));

@sum(OperadorCelda(i,j):CTcost(j)*NTij(i,j))+@sum(ConjuntoTotal(i,j,k):Scost(
j,k)*tijk(i,j,k))<=B;

@for(ConjuntoTotal(i,j,k):@Gin(NSijk));
@for(ConjuntoTotal(i,j,k):@Gin(tijk));
@for(OperadorCelda(i,j):@Bin(NTij));
@for(OperadorCelda(i,j):@Bin(CTij));
@for(ConjuntoTotal(i,j,k):@Bin(Qijk));

@for(OperadorCelda(i,j):NTij(i,j)+Tij(i,j)=CTij(i,j));
@for(OperadorCelda(i,j):NTij(i,j)+Tij(i,j)<=1);

@for(ConjuntoTotal(i,j,k):NSijk(i,j,k)=Sijk(i,j,k)+tijk(i,j,k)+NTij(i,j));
@for(ConjuntoTotal(i,j,k):Sijk(i,j,k)+tijk(i,j,k)+NTij(i,j)<=10);

@for(ConjuntoTotal(i,j,k):tijk(i,j,k)<=9*Qijk(i,j,k));
@for(ConjuntoTotal(i,j,k):tijk(i,j,k)>=Qijk(i,j,k));

@for(ConjuntoTotal(i,j,k):[mINsKILL]
NTij(i,j)+tijk(i,j,k)>=CSjk(j,k)*Qijk(i,j,k)-M*(1-NTij(i,j)));
@for(ConjuntoTotal(i,j,k):[mINsKILL2] sijk(i,j,k)+tijk(i,j,k)>=CSjk(j,k)*(1-
NTij(i,j))-M*(1-Qijk(i,j,k)));

@for(ConjuntoTotal(i,j,k):CTij(i,j)>=Qijk(i,j,k));
```

```
@for(OperadorCelda(i,j):@sum(Estaciones(k)|k#LE#Sj(j):Qijk(i,j,k))>=
NTij(i,j));

CostoTotal =
@sum(OperadorCelda(i,j):CTcost(j)*NTij(i,j))+@sum(ConjuntoTotal(i,j,k):Scost(
j,k)*tijk(i,j,k));
EndSubmodel


Calc:
@IFC(cr#EQ#1:


@Solve(Botella);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','BA')=
ConjuntoTotal,Aijk;

@for(ConjuntoTotal(i,j,k):BAijk(i,j,k) = Aijk(i,j,k));

@Solve(Smin,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','AMin')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSMin') = Operadores, S;
@ODBC('OptModel','SolutionP','U') = U;
@ODBC('OptModel','SolutionP','TSSMin') = TSS;

LB = U;

@Solve(Stotal,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ATot')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSTot') = Operadores, S;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','S')=
ConjuntoTotal,Sijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','P')=
ConjuntoTotal,Pijk;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','Training')=
OperadorCelda,Tij;
@ODBC('OptModel','SolutionP','TSSTot') = TSS;

@Solve(CrossTraining);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ST')=
ConjuntoTotal,tijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','NS')=
ConjuntoTotal,NSijk;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','CellTraining CT')=
OperadorCelda,NTij;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','NewCellTraining')=
OperadorCelda,CTij;
@ODBC('OptModel','SolutionP','Cr Cost') = CostoTotal ;

@for(ConjuntoTotal(i,j,k):@Release(Sijk(i,j,k)));
@for(OperadorCelda(i,j):@Release(Tij(i,j)));

@for(ConjuntoTotal(i,j,k):Sijk(i,j,k) = NSijk(i,j,k));
@for(OperadorCelda(i,j):Tij(i,j) = CTij(i,j));
```

```
@for(ConjuntoTotal(i,j,k):@Release(BAijk(i,j,k)));

@Release(LB);


@Solve(Botella);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','BACr')=
ConjuntoTotal,Aijk;

@for(ConjuntoTotal(i,j,k):BAijk(i,j,k) = Aijk(i,j,k));

@Solve(Smin,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ACrMin')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSCrMin') = Operadores, S;
@ODBC('OptModel','SolutionP','UCr') = U;
@ODBC('OptModel','SolutionP','TSSMinCr') = TSS;

LB = U;

@Solve(Stotal,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ACrTot')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSCrTot') = Operadores, S;
@ODBC('OptModel','SolutionP','TSSTotCr') = TSS;


@ELSE


@Solve(Botella);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','BA')=
ConjuntoTotal,Aijk;

@for(ConjuntoTotal(i,j,k):BAijk(i,j,k) = Aijk(i,j,k));

@Solve(Smin,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','AMin')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSMin') = Operadores, S;
@ODBC('OptModel','SolutionP','U') = U;
@ODBC('OptModel','SolutionP','TSSMin') = TSS;

LB = U;

@Solve(Stotal,MinTotCons);
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ATot')=
ConjuntoTotal,Aijk;
@ODBC('OptModel','SolutionOp','Operator','SSTot') = Operadores, S;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','S')=
ConjuntoTotal,Sijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','P')=
ConjuntoTotal,Pijk;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','Training')=
OperadorCelda,Tij;
@ODBC('OptModel','SolutionP','TSSTot') = TSS;
```

```
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ST')=
ConjuntoTotal,tijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','NS')=
ConjuntoTotal,NSijk;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','CellTraining CT')=
OperadorCelda,NTij;
@ODBC('OptModel','SolutionOpCell','Operator','Cell','NewCellTraining')=
OperadorCelda,CTij;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','BACr')=
ConjuntoTotal,tijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ACrTot')=
ConjuntoTotal,tijk;
@ODBC('OptModel','SolutionOpCellSta','Operator','Cell','Station','ACrMin')=
ConjuntoTotal,tijk;
@ODBC('OptModel','SolutionOp','Operator','SSCrTot') = Operadores, Dummy;
@ODBC('OptModel','SolutionOp','Operator','SSCrMin') = Operadores, Dummy;
@ODBC('OptModel','SolutionP','UCr') = dum2;
@ODBC('OptModel','SolutionP','TSSMinCr') = dum2;
@ODBC('OptModel','SolutionP','TSSTotCr') = dum2;
@ODBC('OptModel','SolutionP','Cr Cost') = dum2 ;

);

EndCalc
END
```

C. Access VBA Code

Appendix C shows the Visual Basic code utilized in the Access Interface created.

```
Public Enum LSerrorCodeLng
  LSERR_NO_ERROR_LNG = 0
  LSERR_OUT_OF_MEMORY_LNG = 1
  LSERR_UNABLE_TO_OPEN_LOG_FILE_LNG = 2
  LSERR_INVALID_NULL_POINTER_LNG = 3
  LSERR_INVALID_INPUT_LNG = 4
End Enum

Public Enum LScallbackInfoCodeLng
  LS_IINFO_VARIABLES_LNG = 0
  LS_IINFO_VARIABLES_INTEGER_LNG = 1
  LS_IINFO_VARIABLES_NONLINEAR_LNG = 2
  LS_IINFO_CONSTRAINTS_LNG = 3
  LS_IINFO_CONSTRAINTS_NONLINEAR_LNG = 4
  LS_IINFO_NONZEROS_LNG = 5
  LS_IINFO_NONZEROS_NONLINEAR_LNG = 6
  LS_IINFO_ITERATIONS_LNG = 7
  LS_IINFO_BRANCHES_LNG = 8
  LS_DINFO_SUMINF_LNG = 9
  LS_DINFO_OBJECTIVE_LNG = 10
  LS_DINFO_MIP_BOUND_LNG = 11
  LS_DINFO_MIP_BEST_OBJECTIVE_LNG = 12
End Enum

Public Enum LSstatusCodeLng
  LS_STATUS_GLOBAL_LNG = 0
  LS_STATUS_INFEASIBLE_LNG = 1
  LS_STATUS_UNBOUNDED_LNG = 2
  LS_STATUS_UNDETERMINED_LNG = 3
  LS_STATUS_FEASIBLE_LNG = 4
  LS_STATUS_INFORUNB_LNG = 5
  LS_STATUS_LOCAL_LNG = 6
  LS_STATUS_LOCAL_INFEASIBLE_LNG = 7
  LS_STATUS_CUTOFF_LNG = 8
  LS_STATUS_NUMERIC_ERROR_LNG = 9
End Enum

Public Declare Function LSclearPointersLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long)

Public Declare Function LScloseLogFileLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long) As Long
```

```
Public Declare Function LScreateEnvLng _
Lib "LINGD11.DLL" () As Long

Public Declare Function LSdeleteEnvLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long) As Long

Public Declare Function LSexecuteScriptLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long, ByVal cScript As String) As Long

Public Declare Function LSgetCallbackInfoDoubleLng _
Lib "LINGD11.DLL" Alias "LSgetCallbackInfoLng" (ByVal pLINGO As Long, _
ByVal nObject As Long, ByRef dResult As Double) As Long

Public Declare Function LSgetCallbackInfoLongLng _
Lib "LINGD11.DLL" Alias "LSgetCallbackInfoLng" (ByVal pLINGO As Long, _
ByVal nObject As Long, ByRef nResult As Long) As Long

Public Declare Function LSopenLogFileLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long, ByVal cFname As String) As Long

Public Declare Function LSsetCallbackErrorLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long, ByVal pcbf As Long, _
ByRef pUserData As Double) As Long

Public Declare Function LSsetCallbackSolverLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long, ByVal pcbf As Long, _
ByRef pUserData As Double) As Long

Public Declare Function LSsetPointerLng _
Lib "LINGD11.DLL" (ByVal pLINGO As Long, ByRef dObject As Double, _
ByRef nPointersNow As Long) As Long

Function Solve()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim sqlTxt As String

' Create the LINGO environment object
   Dim pLINGO As Long
   pLINGO = LScreateEnvLng()
   If pLINGO = 0 Then
     MsgBox ("Unable to create LINGO Environment.")
     GoTo FinalExit
   End If
```

```
' Open LINGO's log file
  Dim nError As Long
  nError = LSopenLogFileLng(pLINGO, CurrentProject.Path & "\LINGO.log")

  If nError <> 0 Then GoTo ErrorExit

 ' Load the Lingo script from  Table
  Dim cScript As String

  Set dbMaindataBase = CurrentDb
  sqlTxt = "SELECT LingoScript.Script FROM  [LingoScript]"
  Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)

  Dim n As Integer

  rstDataBase.MoveLast
  n = rstDataBase.RecordCount
  rstDataBase.MoveFirst

  Dim i As Integer
  For i = 1 To n
    cScript = cScript & rstDataBase.Fields("Script").Value & Chr(10)
    rstDataBase.MoveNext
  Next i
  cScript = cScript & Chr(0) ' End script with a null byte

' Run the script
  nError = LSexecuteScriptLng(pLINGO, cScript)

' Close the log file
  LScloseLogFileLng (pLINGO)

' Problems?
  If nError <> 0 Or _
   dStatus <> LS_STATUS_GLOBAL_LNG Then
    MsgBox ("Unable to solve!")
    GoTo ErrorExit
  End If

  GoTo FinalExit

ErrorExit:
  MsgBox ("LINGO Error Code: " & nError&)

FinalExit:
  LSdeleteEnvLng (pLINGO)
```

```
   MsgBox ("Model has executed successfully")
     End Function

Function SolP()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim sqlTxt As String


Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT SolutionP.* FROM  [SolutionP] "

Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
rstDataBase.MoveFirst

            With rstDataBase
            .Edit
            .Fields("U") = 0
            .Fields("UCr") = 0
            .Fields("TSSMin") = 0
            .Fields("TSSTot") = 0
            .Fields("TSSMinCr") = 0
            .Fields("TSSTotCr") = 0
            .Fields("Cr Cost") = 0
            .Update
            End With


End Function
Function Sta()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim qtyCells As Single


Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT Stations.* FROM  [Stations] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
```

```
Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)

qtyCells = rstCells(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM Stations"
        DoCmd.SetWarnings True

        For i = 1 To qtyCells
            With rstDataBase
            .AddNew
            .Fields("Cell") = i
            .Update
            End With
        Next i

End Function
Function CellPa()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim qtyCells As Single


Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT CellP.* FROM  [CellP] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "

Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)

qtyCells = rstCells(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM CellP"
        DoCmd.SetWarnings True

        For i = 1 To qtyCells
            With rstDataBase
            .AddNew
            .Fields("Cell") = i
```

```
            .Update
            End With
        Next i

End Function
Function OpCellPa()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim rstOps As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim ops As String
 Dim qtyCells As Single
 Dim qtyOps As Single

Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT OpCellP.* FROM  [OpCellP] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
ops = "SELECT Parameters.Operators FROM  [Parameters] "


Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)
Set rstOps = dbMaindataBase.OpenRecordset(ops)


qtyCells = rstCells(0)
qtyOps = rstOps(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM OpCellP"
        DoCmd.SetWarnings True

        For i = 1 To qtyOps
            For j = 1 To qtyCells
                With rstDataBase
                .AddNew
                .Fields("Operator") = i
                .Fields("Cell") = j
                .Update
                End With
            Next j
        Next i
```

```
End Function
Function CellStaPa()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim rstSta As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim stat As String
 Dim qtyCells As Single

Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT CellStaP.* FROM  [CellStaP] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
stat = "SELECT Stations.* FROM  [Stations] "


Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)
Set rstSta = dbMaindataBase.OpenRecordset(stat)

qtyCells = rstCells(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM CellStaP"
        DoCmd.SetWarnings True

        rstSta.MoveFirst
        For j = 1 To qtyCells
            For k = 1 To rstSta("Stations").Value
                With rstDataBase
                .AddNew
                .Fields("Cell") = j
                .Fields("Station") = k
                .Update
                End With
            Next k
            rstSta.MoveNext
        Next j

End Function
Function OpCellStaPa()
```

```
 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim rstOps As DAO.Recordset
 Dim rstSta As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim stat As String
 Dim ops As String
 Dim qtyCells As Single
 Dim qtyOps As Single

Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT OpCellStaP.* FROM  [OpCellStaP] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
stat = "SELECT Stations.* FROM  [Stations] "
ops = "SELECT Parameters.Operators FROM  [Parameters] "


Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)
Set rstSta = dbMaindataBase.OpenRecordset(stat)
Set rstOps = dbMaindataBase.OpenRecordset(ops)


qtyCells = rstCells(0)
qtyOps = rstOps(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM OpCellStaP"
        DoCmd.SetWarnings True

        For i = 1 To qtyOps
            rstSta.MoveFirst
            For j = 1 To qtyCells
                For k = 1 To rstSta("Stations").Value
                    With rstDataBase
                    .AddNew
                    .Fields("Operator") = i
                    .Fields("Cell") = j
                    .Fields("Station") = k
                    .Update
                    End With
                Next k
                rstSta.MoveNext
```

```
        Next j
      Next i

End Function
Function SolOpCellSta()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim rstOps As DAO.Recordset
 Dim rstSta As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim stat As String
 Dim ops As String
 Dim qtyCells As Single
 Dim qtyOps As Single

Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT SolutionOpCellSta.* FROM  [SolutionOpCellSta] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
stat = "SELECT Stations.* FROM  [Stations] "
ops = "SELECT Parameters.Operators FROM  [Parameters] "


Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)
Set rstSta = dbMaindataBase.OpenRecordset(stat)
Set rstOps = dbMaindataBase.OpenRecordset(ops)


qtyCells = rstCells(0)
qtyOps = rstOps(0)

      DoCmd.SetWarnings False
      DoCmd.RunSQL "DELETE * FROM SolutionOpCellSta"
      DoCmd.SetWarnings True

      For i = 1 To qtyOps
        rstSta.MoveFirst
        For j = 1 To qtyCells
          For k = 1 To rstSta("Stations").Value
            With rstDataBase
            .AddNew
            .Fields("Operator") = i
```

```
                .Fields("Cell") = j
                .Fields("Station") = k
                .Update
                End With
            Next k
            rstSta.MoveNext
        Next j
      Next i

End Function
Function SolOpCell()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstCells As DAO.Recordset
 Dim rstOps As DAO.Recordset
 Dim cells As String
 Dim sqlTxt As String
 Dim ops As String
 Dim qtyCells As Single
 Dim qtyOps As Single

Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT SolutionOpCell.* FROM  [SolutionOpCell] "
cells = "SELECT Parameters.Cells FROM  [Parameters] "
ops = "SELECT Parameters.Operators FROM  [Parameters] "


Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstCells = dbMaindataBase.OpenRecordset(cells)
Set rstOps = dbMaindataBase.OpenRecordset(ops)


qtyCells = rstCells(0)
qtyOps = rstOps(0)

        DoCmd.SetWarnings False
        DoCmd.RunSQL "DELETE * FROM SolutionOpCell"
        DoCmd.SetWarnings True

        For i = 1 To qtyOps
            For j = 1 To qtyCells
                With rstDataBase
                .AddNew
                .Fields("Operator") = i
```

```
            .Fields("Cell") = j
            .Update
            End With
         Next j
      Next i

End Function
Function SolOp()

 Dim dbMaindataBase As DAO.Database
 Dim rstDataBase As DAO.Recordset
 Dim rstOps As DAO.Recordset
 Dim ops As String
 Dim sqlTxt As String
 Dim qtyOps As Single


Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT SolutionOp.* FROM  [SolutionOp] "
ops = "SELECT Parameters.Operators FROM  [Parameters] "

Set rstDataBase = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstOps = dbMaindataBase.OpenRecordset(ops)

qtyOps = rstOps(0)
rstDataBase.MoveLast

      DoCmd.SetWarnings False
      DoCmd.RunSQL "DELETE * FROM SolutionOp"
      DoCmd.SetWarnings True

      For i = 1 To qtyOps
         With rstDataBase
         .AddNew
         .Fields("Operator") = i
         .Update
         End With
      Next i

End Function
Function SkillFix()

 Dim dbMaindataBase As DAO.Database
 Dim rstT As DAO.Recordset
 Dim rstS As DAO.Recordset
```

```
 Dim record As DAO.Recordset
 Dim sqlTxt As String
 Dim sqlTxt2 As String

 Set dbMaindataBase = CurrentDb

sqlTxt = "SELECT OpCellP.* FROM  [OpCellP] "
sqlTxt2 = "SELECT OpCellStaP.* FROM  [OpCellStaP] "

Set rstT = dbMaindataBase.OpenRecordset(sqlTxt)
Set rstS = dbMaindataBase.OpenRecordset(sqlTxt2)
rstT.MoveLast
rstS.MoveLast
nT = rstT.RecordCount
nS = rstS.RecordCount

        rstT.MoveFirst
        For k = 1 To nT

            i = rstT.Fields("Operator").Value
            j = rstT.Fields("Cell").Value

            rstS.MoveFirst
            For d = 1 To nS
                If rstS.Fields("Operator").Value = i And rstS.Fields("Cell").Value = j Then
                    If rstT.Fields("Cell Training").Value = False And rstS.Fields("Skill").Value <> 0
Then
                        rstS.Edit
                        rstS.Fields("Skill").Value = 0
                        rstS.Update
                    ElseIf rstT.Fields("Cell Training").Value = True And rstS.Fields("Skill").Value = 0
Then
                        rstS.Edit
                        rstS.Fields("Skill").Value = 5
                        rstS.Update
                    End If
                End If
                rstS.MoveNext
            Next d
            rstT.MoveNext
        Next k

End Function
Function ExportResults()
```

```
Dim db As Database, rs As Recordset
Dim i As Integer
Dim Op As String, OpCel As String, OpCellSta As String, Pa As String

Dim xlApp As Object
Set xlApp = CreateObject("Excel.Application")

xlApp.Workbooks.Add
xlApp.Visible = True

xlApp.Application.Screenupdating = False

Set db = CurrentDb

Op = "SELECT SolutionOp.* FROM SolutionOp"
OpCell = "SELECT SolutionOpCell.* FROM SolutionOpCell"
OpCellSta = "SELECT SolutionOpCellSta.* FROM SolutionOpCellSta"
Pa = "SELECT SolutionP.* FROM SolutionP"

Set rs = db.OpenRecordset(OpCellSta)

        For i = 0 To rs.Fields.Count - 1
            xlApp.Sheets(1).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

         xlApp.Sheets(1).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

         xlApp.Sheets(1).Range("A1").Select

         xlApp.Sheets(1).Name = "OperatorCellStationResults "

         xlApp.Sheets(2).Activate


 Set rs = db.OpenRecordset(OpCell)

        For i = 0 To rs.Fields.Count - 1
            xlApp.Sheets(2).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

         xlApp.Sheets(2).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

         xlApp.Sheets(2).Range("A1").Select

         xlApp.Sheets(2).Name = "OperatorCellResults "
```

```
        xlApp.Sheets(3).Activate

    Set rs = db.OpenRecordset(Op)

        For i = 0 To rs.Fields.Count - 1
            xlApp.Sheets(3).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

        xlApp.Sheets(3).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

        xlApp.Sheets(3).Name = "OperatorResults"

        xlApp.Sheets.Add After:=xlApp.Sheets(xlApp.Sheets.Count)

        xlApp.Sheets(4).Name = "Totals Results"

        xlApp.Sheets(4).Activate


     Set rs = db.OpenRecordset(Pa)

      For i = 0 To rs.Fields.Count - 1
          xlApp.Sheets(4).cells(1, i + 1).Value = rs.Fields(i).Name
      Next

      xlApp.Sheets(4).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

      'xlApp.Application.ActiveWorkbook.SaveAs FileName:=CurrentProject.Path &
"\Results.xlsx"

      xlApp.Sheets(1).Activate
      xlApp.Application.Screenupdating = True

    Set xlApp = Nothing
    Set rs = Nothing
    Set db = Nothing

End Function
Function ExportParameters()

    Dim db As Database, rs As Recordset
    Dim i As Integer
    Dim Cell As String, OpCel As String, OpCellSta As String, Pa As String, CellSta As String

    Dim xlApp As Object
```

```
Set xlApp = CreateObject("Excel.Application")


 xlApp.Workbooks.Add
 xlApp.Visible = True

 xlApp.Application.Screenupdating = False


Set db = CurrentDb

Cell = "SELECT CellP.* FROM CellP"
OpCell = "SELECT OpCellP.* FROM OpCellP"
OpCellSta = "SELECT OpCellStaP.* FROM OpCellStaP"
CellSta = "SELECT CellStaP.* FROM CellStaP"
Pa = "SELECT Parameters.* FROM [Parameters]"


    Set rs = db.OpenRecordset(OpCellSta)

        For i = 0 To rs.Fields.Count - 1
            xlApp.Sheets(1).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

         xlApp.Sheets(1).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

         xlApp.Sheets(1).Range("A1").Select

         xlApp.Sheets(1).Name = "OperatorCellStationSettings "

         xlApp.Sheets(2).Activate


 Set rs = db.OpenRecordset(OpCell)

        For i = 0 To rs.Fields.Count - 1
            xlApp.Sheets(2).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

         xlApp.Sheets(2).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

         xlApp.Sheets(2).Range("A1").Select

         xlApp.Sheets(2).Name = "OperatorCellSettings "
```

```
        xlApp.Sheets(3).Activate

   Set rs = db.OpenRecordset(CellSta)

        For i = 0 To rs.Fields.Count - 1
           xlApp.Sheets(3).cells(1, i + 1).Value = rs.Fields(i).Name
        Next

        xlApp.Sheets(3).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

        xlApp.Sheets(3).Name = "CellStationSettings"

        xlApp.Sheets.Add After:=xlApp.Sheets(xlApp.Sheets.Count)

        xlApp.Sheets(4).Name = "CellSettings"

        xlApp.Sheets(4).Activate


    Set rs = db.OpenRecordset(Cell)

      For i = 0 To rs.Fields.Count - 1
         xlApp.Sheets(4).cells(1, i + 1).Value = rs.Fields(i).Name
      Next

      xlApp.Sheets(4).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

        xlApp.Sheets.Add After:=xlApp.Sheets(xlApp.Sheets.Count)

        xlApp.Sheets(5).Name = "Settings"

        xlApp.Sheets(5).Activate

    Set rs = db.OpenRecordset(Pa)

      For i = 0 To rs.Fields.Count - 1
         xlApp.Sheets(5).cells(1, i + 1).Value = rs.Fields(i).Name
      Next

      xlApp.Sheets(5).cells(1, 1).Offset(1, 0).CopyFromRecordset rs

      'xlApp.Application.ActiveWorkbook.SaveAs FileName:=CurrentProject.Path &
"\Results.xlsx"

      xlApp.Sheets(1).Activate
      xlApp.Application.Screenupdating = True
```

```
    Set xlApp = Nothing
    Set rs = Nothing
    Set db = Nothing

End Function

Private Declare Function IsWindowVisible Lib "user32" (ByVal hwnd As Long) As Long
Dim dwReturn As Long

Const SW_HIDE = 0
Const SW_SHOWNORMAL = 1
Const SW_SHOWMINIMIZED = 2
Const SW_SHOWMAXIMIZED = 3

Private Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, _
     ByVal nCmdShow As Long) As Long

     Public Function fAccessWindow(Optional Procedure As String, Optional SwitchStatus As
Boolean, Optional StatusCheck As Boolean) As Boolean
If Procedure = "Hide" Then
    dwReturn = ShowWindow(Application.hWndAccessApp, SW_HIDE)
End If
If Procedure = "Show" Then
    dwReturn = ShowWindow(Application.hWndAccessApp, SW_SHOWMAXIMIZED)
End If
If Procedure = "Minimize" Then
    dwReturn = ShowWindow(Application.hWndAccessApp, SW_SHOWMINIMIZED)
End If
If SwitchStatus = True Then
    If IsWindowVisible(hWndAccessApp) = 1 Then
        dwReturn = ShowWindow(Application.hWndAccessApp, SW_HIDE)
    Else
        dwReturn = ShowWindow(Application.hWndAccessApp, SW_SHOWMAXIMIZED)
    End If
End If
If StatusCheck = True Then
    If IsWindowVisible(hWndAccessApp) = 0 Then
        fAccessWindow = False
    End If
    If IsWindowVisible(hWndAccessApp) = 1 Then
        fAccessWindow = True
    End If
End If
End Function
```

D. Case Study Parameters Tables

Appendix D shows tables with all the parameters utilized in the case study created in this work.

Table 23. Case study preferences and skills for each operator and cell station

Operator	Cell	Station	Preference	Skill
1	1	1	5	5
1	1	2	8	7
1	1	3	10	6
1	1	4	5	7
1	1	5	10	1
1	1	6	7	8
1	2	1	6	8
1	2	2	7	3
1	2	3	8	8
1	2	4	10	4
1	3	1	3	7
1	3	2	4	7
1	3	3	7	2
1	3	4	10	6
1	3	5	1	5
1	3	6	4	2
1	3	7	6	5
1	3	8	8	6
1	3	9	7	7
1	4	1	8	8
1	4	2	5	3
1	4	3	9	1
1	4	4	1	4
1	4	5	2	8
1	4	6	4	8
1	4	7	10	7
1	4	8	5	5
1	5	1	4	7
1	5	2	1	7
1	5	3	8	6
1	5	4	6	1
1	5	5	6	7
1	5	6	4	6
1	5	7	2	8
2	1	1	4	6
2	1	2	8	2
2	1	3	6	7

Operator	Cell	Station	Preference	Skill
2	1	4	1	7
2	1	5	9	7
2	1	6	7	7
2	2	1	7	1
2	2	2	2	5
2	2	3	6	7
2	2	4	5	7
2	3	1	8	5
2	3	2	9	7
2	3	3	1	8
2	3	4	1	1
2	3	5	8	4
2	3	6	4	8
2	3	7	6	8
2	3	8	8	8
2	3	9	3	7
2	4	1	10	7
2	4	2	5	6
2	4	3	8	6
2	4	4	3	8
2	4	5	5	3
2	4	6	4	6
2	4	7	5	6
2	4	8	8	4
2	5	1	5	5
2	5	2	7	6
2	5	3	4	3
2	5	4	1	6
2	5	5	1	3
2	5	6	6	8
2	5	7	10	6
3	1	1	5	5
3	1	2	10	7
3	1	3	5	2
3	1	4	9	6
3	1	5	3	7
3	1	6	8	6
3	2	1	5	6
3	2	2	5	7
3	2	3	10	8
3	2	4	1	9
3	3	1	3	7

Operator	Cell	Station	Preference	Skill
3	3	2	4	7
3	3	3	3	7
3	3	4	6	8
3	3	5	3	7
3	3	6	2	3
3	3	7	7	4
3	3	8	10	5
3	3	9	2	7
3	4	1	10	5
3	4	2	7	7
3	4	3	6	3
3	4	4	2	7
3	4	5	4	5
3	4	6	8	8
3	4	7	1	6
3	4	8	1	2
3	5	1	3	4
3	5	2	4	2
3	5	3	8	5
3	5	4	6	6
3	5	5	10	6
3	5	6	6	5
3	5	7	6	6
4	1	1	9	7
4	1	2	6	6
4	1	3	2	4
4	1	4	9	5
4	1	5	5	6
4	1	6	2	1
4	2	1	6	5
4	2	2	6	5
4	2	3	5	7
4	2	4	6	8
4	3	1	1	5
4	3	2	10	5
4	3	3	8	6
4	3	4	4	2
4	3	5	6	6
4	3	6	2	5
4	3	7	2	5
4	3	8	1	7
4	3	9	9	5

Operator	Cell	Station	Preference	Skill
4	4	1	10	4
4	4	2	7	6
4	4	3	7	7
4	4	4	6	1
4	4	5	10	3
4	4	6	8	7
4	4	7	2	4
4	4	8	10	7
4	5	1	8	1
4	5	2	10	1
4	5	3	1	4
4	5	4	1	6
4	5	5	4	6
4	5	6	8	1
4	5	7	7	2
5	1	1	2	4
5	1	2	2	5
5	1	3	5	6
5	1	4	1	7
5	1	5	4	5
5	1	6	4	6
5	2	1	8	0
5	2	2	1	0
5	2	3	8	0
5	2	4	10	0
5	3	1	2	1
5	3	2	4	5
5	3	3	9	7
5	3	4	1	6
5	3	5	9	8
5	3	6	6	4
5	3	7	1	7
5	3	8	7	7
5	3	9	10	4
5	4	1	5	0
5	4	2	1	0
5	4	3	3	0
5	4	4	2	0
5	4	5	7	0
5	4	6	1	0
5	4	7	6	0
5	4	8	4	0

Operator	Cell	Station	Preference	Skill
5	5	1	8	0
5	5	2	10	0
5	5	3	5	0
5	5	4	2	0
5	5	5	10	0
5	5	6	8	0
5	5	7	1	0
6	1	1	6	0
6	1	2	4	0
6	1	3	6	0
6	1	4	8	0
6	1	5	4	0
6	1	6	5	0
6	2	1	4	3
6	2	2	2	5
6	2	3	10	7
6	2	4	9	6
6	3	1	8	0
6	3	2	1	0
6	3	3	6	0
6	3	4	1	0
6	3	5	10	0
6	3	6	6	0
6	3	7	7	0
6	3	8	5	0
6	3	9	4	0
6	4	1	9	1
6	4	2	10	7
6	4	3	5	6
6	4	4	4	5
6	4	5	5	6
6	4	6	9	1
6	4	7	10	7
6	4	8	5	6
6	5	1	3	0
6	5	2	6	0
6	5	3	4	0
6	5	4	9	0
6	5	5	8	0
6	5	6	6	0
6	5	7	10	0
7	1	1	7	0

Operator	Cell	Station	Preference	Skill
7	1	2	7	0
7	1	3	6	0
7	1	4	8	0
7	1	5	1	0
7	1	6	3	0
7	2	1	6	0
7	2	2	4	0
7	2	3	6	0
7	2	4	7	0
7	3	1	4	6
7	3	2	7	4
7	3	3	10	7
7	3	4	3	3
7	3	5	10	4
7	3	6	9	5
7	3	7	6	2
7	3	8	10	6
7	3	9	7	7
7	4	1	3	0
7	4	2	3	0
7	4	3	8	0
7	4	4	6	0
7	4	5	9	0
7	4	6	7	0
7	4	7	7	0
7	4	8	3	0
7	5	1	10	7
7	5	2	10	1
7	5	3	10	5
7	5	4	6	7
7	5	5	1	6
7	5	6	7	7
7	5	7	8	1
8	1	1	6	0
8	1	2	7	0
8	1	3	8	0
8	1	4	9	0
8	1	5	1	0
8	1	6	10	0
8	2	1	4	7
8	2	2	2	5
8	2	3	2	5

Operator	Cell	Station	Preference	Skill
8	2	4	8	7
8	3	1	3	0
8	3	2	7	0
8	3	3	1	0
8	3	4	4	0
8	3	5	8	0
8	3	6	1	0
8	3	7	4	0
8	3	8	4	0
8	3	9	7	0
8	4	1	5	5
8	4	2	4	7
8	4	3	2	2
8	4	4	10	2
8	4	5	4	7
8	4	6	3	8
8	4	7	5	7
8	4	8	1	5
8	5	1	4	0
8	5	2	3	0
8	5	3	5	0
8	5	4	2	0
8	5	5	6	0
8	5	6	5	0
8	5	7	1	0
9	1	1	3	6
9	1	2	3	8
9	1	3	10	7
9	1	4	6	5
9	1	5	6	5
9	1	6	6	7
9	2	1	3	0
9	2	2	3	0
9	2	3	9	0
9	2	4	4	0
9	3	1	7	0
9	3	2	6	0
9	3	3	1	0
9	3	4	3	0
9	3	5	6	0
9	3	6	5	0
9	3	7	8	0

Operator	Cell	Station	Preference	Skill
9	3	8	3	0
9	3	9	1	0
9	4	1	8	0
9	4	2	2	0
9	4	3	8	0
9	4	4	3	0
9	4	5	7	0
9	4	6	10	0
9	4	7	9	0
9	4	8	6	0
9	5	1	8	0
9	5	2	10	0
9	5	3	5	0
9	5	4	3	0
9	5	5	9	0
9	5	6	9	0
9	5	7	4	0
10	1	1	7	0
10	1	2	5	0
10	1	3	9	0
10	1	4	3	0
10	1	5	8	0
10	1	6	9	0
10	2	1	1	7
10	2	2	10	5
10	2	3	5	6
10	2	4	6	4
10	3	1	2	0
10	3	2	9	0
10	3	3	9	0
10	3	4	2	0
10	3	5	3	0
10	3	6	5	0
10	3	7	10	0
10	3	8	1	0
10	3	9	2	0
10	4	1	3	0
10	4	2	8	0
10	4	3	1	0
10	4	4	2	0
10	4	5	2	0
10	4	6	10	0

Operator	Cell	Station	Preference	Skill
10	4	7	9	0
10	4	8	3	0
10	5	1	9	0
10	5	2	7	0
10	5	3	2	0
10	5	4	10	0
10	5	5	2	0
10	5	6	5	0
10	5	7	3	0
11	1	1	1	0
11	1	2	6	0
11	1	3	7	0
11	1	4	2	0
11	1	5	10	0
11	1	6	8	0
11	2	1	10	0
11	2	2	3	0
11	2	3	5	0
11	2	4	1	0
11	3	1	10	5
11	3	2	4	6
11	3	3	6	7
11	3	4	10	7
11	3	5	10	5
11	3	6	1	1
11	3	7	5	7
11	3	8	8	6
11	3	9	9	7
11	4	1	10	0
11	4	2	4	0
11	4	3	5	0
11	4	4	5	0
11	4	5	4	0
11	4	6	8	0
11	4	7	9	0
11	4	8	3	0
11	5	1	6	0
11	5	2	9	0
11	5	3	7	0
11	5	4	2	0
11	5	5	9	0
11	5	6	5	0

Operator	Cell	Station	Preference	Skill
11	5	7	2	0
12	1	1	10	0
12	1	2	8	0
12	1	3	6	0
12	1	4	6	0
12	1	5	2	0
12	1	6	9	0
12	2	1	6	0
12	2	2	10	0
12	2	3	1	0
12	2	4	6	0
12	3	1	2	0
12	3	2	3	0
12	3	3	6	0
12	3	4	2	0
12	3	5	3	0
12	3	6	7	0
12	3	7	4	0
12	3	8	4	0
12	3	9	6	0
12	4	1	1	1
12	4	2	1	3
12	4	3	9	6
12	4	4	6	7
12	4	5	2	7
12	4	6	8	8
12	4	7	6	5
12	4	8	7	5
12	5	1	9	0
12	5	2	9	0
12	5	3	6	0
12	5	4	4	0
12	5	5	6	0
12	5	6	5	0
12	5	7	1	0
13	1	1	7	0
13	1	2	4	0
13	1	3	2	0
13	1	4	8	0
13	1	5	3	0
13	1	6	1	0
13	2	1	10	0

Operator	Cell	Station	Preference	Skill
13	2	2	5	0
13	2	3	1	0
13	2	4	2	0
13	3	1	1	0
13	3	2	3	0
13	3	3	2	0
13	3	4	5	0
13	3	5	7	0
13	3	6	4	0
13	3	7	7	0
13	3	8	1	0
13	3	9	3	0
13	4	1	10	0
13	4	2	8	0
13	4	3	5	0
13	4	4	6	0
13	4	5	2	0
13	4	6	7	0
13	4	7	6	0
13	4	8	1	0
13	5	1	2	5
13	5	2	3	3
13	5	3	4	6
13	5	4	10	6
13	5	5	4	6
13	5	6	3	3
13	5	7	8	6
14	1	1	3	0
14	1	2	5	0
14	1	3	1	0
14	1	4	8	0
14	1	5	6	0
14	1	6	7	0
14	2	1	5	6
14	2	2	4	8
14	2	3	3	1
14	2	4	6	7
14	3	1	2	0
14	3	2	6	0
14	3	3	5	0
14	3	4	9	0
14	3	5	9	0

Operator	Cell	Station	Preference	Skill
14	3	6	1	0
14	3	7	6	0
14	3	8	7	0
14	3	9	10	0
14	4	1	6	6
14	4	2	7	6
14	4	3	5	2
14	4	4	3	8
14	4	5	4	8
14	4	6	6	7
14	4	7	5	6
14	4	8	9	5
14	5	1	8	7
14	5	2	1	6
14	5	3	2	5
14	5	4	5	6
14	5	5	4	2
14	5	6	7	4
14	5	7	8	6
15	1	1	10	6
15	1	2	9	5
15	1	3	2	7
15	1	4	10	6
15	1	5	5	7
15	1	6	7	7
15	2	1	3	1
15	2	2	9	8
15	2	3	8	8
15	2	4	3	6
15	3	1	3	0
15	3	2	9	0
15	3	3	5	0
15	3	4	7	0
15	3	5	5	0
15	3	6	3	0
15	3	7	9	0
15	3	8	6	0
15	3	9	2	0
15	4	1	2	5
15	4	2	9	7
15	4	3	8	8
15	4	4	9	6

Operator	Cell	Station	Preference	Skill
15	4	5	7	8
15	4	6	7	4
15	4	7	9	8
15	4	8	10	6
15	5	1	3	0
15	5	2	9	0
15	5	3	3	0
15	5	4	4	0
15	5	5	3	0
15	5	6	1	0
15	5	7	4	0
16	1	1	6	7
16	1	2	4	5
16	1	3	2	4
16	1	4	4	7
16	1	5	6	6
16	1	6	5	6
16	2	1	3	0
16	2	2	8	0
16	2	3	5	0
16	2	4	5	0
16	3	1	3	8
16	3	2	9	8
16	3	3	4	7
16	3	4	3	5
16	3	5	8	6
16	3	6	5	5
16	3	7	4	6
16	3	8	5	8
16	3	9	10	6
16	4	1	10	7
16	4	2	8	3
16	4	3	9	5
16	4	4	5	7
16	4	5	9	1
16	4	6	8	6
16	4	7	10	7
16	4	8	2	5
16	5	1	5	0
16	5	2	5	0
16	5	3	10	0
16	5	4	3	0

Operator	Cell	Station	Preference	Skill
16	5	5	9	0
16	5	6	8	0
16	5	7	1	0
17	1	1	9	0
17	1	2	7	0
17	1	3	2	0
17	1	4	10	0
17	1	5	7	0
17	1	6	6	0
17	2	1	6	6
17	2	2	1	5
17	2	3	3	8
17	2	4	6	6
17	3	1	2	7
17	3	2	10	6
17	3	3	1	9
17	3	4	7	5
17	3	5	1	7
17	3	6	5	3
17	3	7	6	5
17	3	8	8	8
17	3	9	10	5
17	4	1	2	0
17	4	2	6	0
17	4	3	1	0
17	4	4	1	0
17	4	5	1	0
17	4	6	4	0
17	4	7	4	0
17	4	8	9	0
17	5	1	9	2
17	5	2	6	5
17	5	3	2	4
17	5	4	2	7
17	5	5	8	6
17	5	6	5	7
17	5	7	3	3
18	1	1	2	5
18	1	2	3	6
18	1	3	10	2
18	1	4	5	7
18	1	5	2	8

Operator	Cell	Station	Preference	Skill
18	1	6	7	7
18	2	1	1	0
18	2	2	6	0
18	2	3	3	0
18	2	4	5	0
18	3	1	8	7
18	3	2	10	9
18	3	3	1	5
18	3	4	6	6
18	3	5	7	1
18	3	6	6	2
18	3	7	5	7
18	3	8	1	7
18	3	9	7	8
18	4	1	4	5
18	4	2	1	3
18	4	3	1	2
18	4	4	7	7
18	4	5	7	7
18	4	6	3	8
18	4	7	4	6
18	4	8	10	5
18	5	1	4	4
18	5	2	1	6
18	5	3	3	7
18	5	4	10	6
18	5	5	1	5
18	5	6	2	7
18	5	7	5	3
19	1	1	8	5
19	1	2	3	7
19	1	3	10	7
19	1	4	2	4
19	1	5	5	7
19	1	6	10	1
19	2	1	7	6
19	2	2	7	4
19	2	3	4	7
19	2	4	8	5
19	3	1	10	0
19	3	2	5	0
19	3	3	6	0

Operator	Cell	Station	Preference	Skill
19	3	4	8	0
19	3	5	1	0
19	3	6	8	0
19	3	7	4	0
19	3	8	8	0
19	3	9	5	0
19	4	1	10	8
19	4	2	1	4
19	4	3	10	6
19	4	4	5	2
19	4	5	5	7
19	4	6	9	6
19	4	7	8	5
19	4	8	1	5
19	5	1	1	5
19	5	2	9	2
19	5	3	7	7
19	5	4	5	6
19	5	5	2	5
19	5	6	2	4
19	5	7	8	7
20	1	1	10	7
20	1	2	8	4
20	1	3	3	7
20	1	4	10	7
20	1	5	5	6
20	1	6	1	8
20	2	1	1	3
20	2	2	8	3
20	2	3	6	7
20	2	4	7	5
20	3	1	8	7
20	3	2	9	6
20	3	3	10	7
20	3	4	7	7
20	3	5	9	6
20	3	6	5	5
20	3	7	6	2
20	3	8	4	7
20	3	9	1	6
20	4	1	3	0
20	4	2	7	0

Operator	Cell	Station	Preference	Skill
20	4	3	4	0
20	4	4	2	0
20	4	5	6	0
20	4	6	8	0
20	4	7	2	0
20	4	8	3	0
20	5	1	2	7
20	5	2	9	2
20	5	3	6	4
20	5	4	10	9
20	5	5	3	5
20	5	6	6	6
20	5	7	3	4
21	1	1	2	3
21	1	2	4	6
21	1	3	9	5
21	1	4	3	7
21	1	5	9	6
21	1	6	3	4
21	2	1	2	2
21	2	2	1	8
21	2	3	1	5
21	2	4	2	6
21	3	1	4	4
21	3	2	2	7
21	3	3	8	5
21	3	4	5	6
21	3	5	10	6
21	3	6	8	7
21	3	7	7	5
21	3	8	4	7
21	3	9	10	5
21	4	1	3	6
21	4	2	6	9
21	4	3	5	5
21	4	4	4	4
21	4	5	3	8
21	4	6	5	3
21	4	7	5	8
21	4	8	8	6
21	5	1	8	0
21	5	2	9	0

Operator	Cell	Station	Preference	Skill
21	5	3	5	0
21	5	4	1	0
21	5	5	3	0
21	5	6	8	0
21	5	7	1	0
22	1	1	3	0
22	1	2	8	0
22	1	3	7	0
22	1	4	8	0
22	1	5	7	0
22	1	6	8	0
22	2	1	9	0
22	2	2	4	0
22	2	3	7	0
22	2	4	4	0
22	3	1	7	0
22	3	2	1	0
22	3	3	10	0
22	3	4	6	0
22	3	5	1	0
22	3	6	7	0
22	3	7	8	0
22	3	8	3	0
22	3	9	7	0
22	4	1	7	0
22	4	2	4	0
22	4	3	2	0
22	4	4	5	0
22	4	5	1	0
22	4	6	6	0
22	4	7	9	0
22	4	8	10	0
22	5	1	4	7
22	5	2	6	1
22	5	3	1	3
22	5	4	5	7
22	5	5	7	6
22	5	6	9	1
22	5	7	5	8
23	1	1	5	0
23	1	2	6	0
23	1	3	10	0

Operator	Cell	Station	Preference	Skill
23	1	4	10	0
23	1	5	8	0
23	1	6	9	0
23	2	1	4	0
23	2	2	5	0
23	2	3	4	0
23	2	4	9	0
23	3	1	10	0
23	3	2	3	0
23	3	3	1	0
23	3	4	4	0
23	3	5	5	0
23	3	6	2	0
23	3	7	2	0
23	3	8	3	0
23	3	9	9	0
23	4	1	9	3
23	4	2	8	4
23	4	3	2	5
23	4	4	4	8
23	4	5	7	7
23	4	6	5	9
23	4	7	4	7
23	4	8	2	5
23	5	1	2	0
23	5	2	2	0
23	5	3	5	0
23	5	4	3	0
23	5	5	1	0
23	5	6	2	0
23	5	7	9	0
24	1	1	3	0
24	1	2	3	0
24	1	3	7	0
24	1	4	2	0
24	1	5	3	0
24	1	6	8	0
24	2	1	8	0
24	2	2	7	0
24	2	3	4	0
24	2	4	10	0
24	3	1	10	6

Operator	Cell	Station	Preference	Skill
24	3	2	7	9
24	3	3	4	6
24	3	4	5	5
24	3	5	7	7
24	3	6	9	7
24	3	7	1	1
24	3	8	3	8
24	3	9	5	7
24	4	1	8	0
24	4	2	2	0
24	4	3	3	0
24	4	4	8	0
24	4	5	8	0
24	4	6	6	0
24	4	7	5	0
24	4	8	7	0
24	5	1	9	0
24	5	2	5	0
24	5	3	9	0
24	5	4	9	0
24	5	5	3	0
24	5	6	4	0
24	5	7	9	0
25	1	1	8	0
25	1	2	10	0
25	1	3	10	0
25	1	4	6	0
25	1	5	4	0
25	1	6	10	0
25	2	1	10	5
25	2	2	3	5
25	2	3	6	9
25	2	4	8	6
25	3	1	5	0
25	3	2	2	0
25	3	3	2	0
25	3	4	3	0
25	3	5	8	0
25	3	6	2	0
25	3	7	2	0
25	3	8	10	0
25	3	9	3	0

Operator	Cell	Station	Preference	Skill
25	4	1	4	0
25	4	2	9	0
25	4	3	6	0
25	4	4	10	0
25	4	5	9	0
25	4	6	2	0
25	4	7	4	0
25	4	8	7	0
25	5	1	7	0
25	5	2	7	0
25	5	3	1	0
25	5	4	1	0
25	5	5	9	0
25	5	6	5	0
25	5	7	3	0
26	1	1	5	8
26	1	2	10	7
26	1	3	6	2
26	1	4	8	6
26	1	5	10	5
26	1	6	5	8
26	2	1	7	0
26	2	2	5	0
26	2	3	6	0
26	2	4	7	0
26	3	1	9	0
26	3	2	8	0
26	3	3	10	0
26	3	4	4	0
26	3	5	5	0
26	3	6	3	0
26	3	7	7	0
26	3	8	7	0
26	3	9	2	0
26	4	1	9	0
26	4	2	8	0
26	4	3	3	0
26	4	4	1	0
26	4	5	9	0
26	4	6	6	0
26	4	7	6	0
26	4	8	4	0

Operator	Cell	Station	Preference	Skill
26	5	1	6	0
26	5	2	6	0
26	5	3	9	0
26	5	4	6	0
26	5	5	6	0
26	5	6	9	0
26	5	7	2	0

Table 24. Case study operator cell training for each cell

Operator	Cell	Cell Training
1	1	TRUE
1	2	TRUE
1	3	TRUE
1	4	TRUE
1	5	TRUE
2	1	TRUE
2	2	TRUE
2	3	TRUE
2	4	TRUE
2	5	TRUE
3	1	TRUE
3	2	TRUE
3	3	TRUE
3	4	TRUE
3	5	TRUE
4	1	TRUE
4	2	TRUE
4	3	TRUE
4	4	TRUE
4	5	TRUE
5	1	TRUE
5	2	FALSE
5	3	TRUE
5	4	FALSE
5	5	FALSE
6	1	FALSE
6	2	TRUE
6	3	FALSE
6	4	TRUE
6	5	FALSE

Operator	Cell	Cell Training
7	1	FALSE
7	2	FALSE
7	3	TRUE
7	4	FALSE
7	5	TRUE
8	1	FALSE
8	2	TRUE
8	3	FALSE
8	4	TRUE
8	5	FALSE
9	1	TRUE
9	2	FALSE
9	3	FALSE
9	4	FALSE
9	5	FALSE
10	1	FALSE
10	2	TRUE
10	3	FALSE
10	4	FALSE
10	5	FALSE
11	1	FALSE
11	2	FALSE
11	3	TRUE
11	4	FALSE
11	5	FALSE
12	1	FALSE
12	2	FALSE
12	3	FALSE
12	4	TRUE
12	5	FALSE
13	1	FALSE
13	2	FALSE
13	3	FALSE
13	4	FALSE
13	5	TRUE
14	1	FALSE
14	2	TRUE
14	3	FALSE
14	4	TRUE
14	5	TRUE
15	1	TRUE
15	2	TRUE

Operator	Cell	Cell Training
15	3	FALSE
15	4	TRUE
15	5	FALSE
16	1	TRUE
16	2	FALSE
16	3	TRUE
16	4	TRUE
16	5	FALSE
17	1	FALSE
17	2	TRUE
17	3	TRUE
17	4	FALSE
17	5	TRUE
18	1	TRUE
18	2	FALSE
18	3	TRUE
18	4	TRUE
18	5	TRUE
19	1	TRUE
19	2	TRUE
19	3	FALSE
19	4	TRUE
19	5	TRUE
20	1	TRUE
20	2	TRUE
20	3	TRUE
20	4	FALSE
20	5	TRUE
21	1	TRUE
21	2	TRUE
21	3	TRUE
21	4	TRUE
21	5	FALSE
22	1	FALSE
22	2	FALSE
22	3	FALSE
22	4	FALSE
22	5	TRUE
23	1	FALSE
23	2	FALSE
23	3	FALSE
23	4	TRUE

Operator	Cell	Cell Training
23	5	FALSE
24	1	FALSE
24	2	FALSE
24	3	TRUE
24	4	FALSE
24	5	FALSE
25	1	FALSE
25	2	TRUE
25	3	FALSE
25	4	FALSE
25	5	FALSE
26	1	TRUE
26	2	FALSE
26	3	FALSE
26	4	FALSE
26	5	FALSE

Table 25. Case study bottlenecks, time requirement, minimum skill, & skill cost for each cell station

Cell	Station	Bottleneck	Time Requirement	Minimum Skill	Training Skill Cost
1	1	FALSE	0.2656	4	11
1	2	FALSE	0.3542	5	13
1	3	FALSE	0.5844	4	10
1	4	TRUE	0.8323	6	8
1	5	FALSE	0.5667	5	15
1	6	FALSE	0.7260	6	14
2	1	FALSE	0.7427	3	12
2	2	TRUE	0.8146	5	8
2	3	FALSE	0.6469	4	8
2	4	FALSE	0.5990	6	14
3	1	FALSE	0.9115	5	15
3	2	FALSE	0.8333	3	12
3	3	FALSE	0.7292	4	12
3	4	FALSE	0.3906	5	9
3	5	FALSE	0.5208	4	11
3	6	FALSE	0.4948	5	13
3	7	FALSE	0.5990	5	8
3	8	TRUE	0.9896	6	14
3	9	FALSE	0.6510	5	13
4	1	TRUE	0.7917	5	14
4	2	FALSE	0.6333	3	12

Cell	Station	Bottleneck	Time Requirement	Minimum Skill	Training Skill Cost
4	3	FALSE	0.5938	5	10
4	4	FALSE	0.6729	4	8
4	5	FALSE	0.6927	7	14
4	6	FALSE	0.6729	6	13
4	7	FALSE	0.6927	6	12
4	8	FALSE	0.3760	5	10
5	1	TRUE	1.0000	4	15
5	2	FALSE	0.8000	5	12
5	3	FALSE	0.6250	3	14
5	4	FALSE	0.4750	6	13
5	5	FALSE	0.8250	6	15
5	6	FALSE	0.7750	4	10
5	7	FALSE	0.9250	5	12

Table 26. Case Study number of stations, minimum assignment and cell training cost for each cell

Cell	Stations per Cell	Minimum Assignment	Cell Training Cost
1	6	0.0625	47
2	4	0.0625	53
3	9	0.0625	50
4	8	0.0625	51
5	7	0.0625	49

Table 27. Case study non-matrix parameters

Operators	Cells	Maximum Cells	Maximum Stations	Cross-training	Cross-training Budget
26	5	1	9	TRUE	1,000