Winter 1-31-2021

Multi-skilled Labor Optimization with Partial Allocation of Resources

Amira Saleh
amiraahmed@aucegypt.edu

The American University in Cairo

School of Sciences and Engineering

MULTI-SKILLED LABOR OPTIMIZATION WITH PARTIAL ALLOCATION OF RESOURCES

A Thesis Submitted to

The Department of Construction Engineering

in partial fulfillment of the requirements for the degree of

Master of Science in Construction Management

By:

Amira Ashraf Saleh

Under the supervision of:

Dr. Ibrahim Abotaleb

Assistant Professor

Department of Construction Engineering

The American University in Cairo

Dr. Ossama Hosny

Professor

Department of Construction Engineering

The American University in Cairo

DEDICATIONS

This work is dedicated to my parents. I would like to express my utmost gratitude to my parents, for their endless support, love, encouragement, and for raising me to believe that anything was possible. I would not be where I am today without you.

Acknowledgement

I would like to express my upmost gratitude to Dr. Ossama Hosny and Dr. Ibrahim Abotaleb for their continuous support and encouragement throughout my studies and research. I would also like to express my appreciation to Dr. Khaled Nassar, Dr. Mohamed Mahdy Marzouk and Dr. Ahmed El-Gendy for having served in my committee. Their comments and questions were greatly valued. I would also like to thank all my professors who have provided me with nothing less than continuous support and encouragement.

ABSTRACT

The current practice of labor allocation in construction schedules assumes single-skilled workforce; meaning that each worker is assumed to be skilled in only one trade. In such practice, at any instance in the project lifecycle, some of the workforce become idle waiting for other labor types to complete their work. Traditionally, companies may relocate idle workers to other projects and return them back to their original project when needed again. This complicates the resource management process and is not often performed successfully, leading to schedule and cost overruns. Alternatively, project managers may keep the idle workforce at their projects because they will be needed at a later stage and pay them in their idle days, which adds unnecessary costs to the project. Another solution would be continuously hiring and laying off labor at need, which has severe negative impacts on projects and firms. Due to the inefficiencies of these solutions, some research discussed the idea of "multi-skilled" labor, where some of the workers may have enough training to carry out different activity types. Multi-skilling decreases inefficiencies and ensures a smooth and continuous progress of works whilst maintaining the workforce and keeping their idle time to a minimum. Multi-skilling could be also used to speed up progress in construction schedules.

Previous research efforts have been made to encourage contractors in pursuing multiskilling as a solution to the non-smooth resource histograms. Yet, the literature falls short in providing a robust multi-skilling framework; specifically, one that considers the cost of training labor and solves the partial allocation problem. The objective of this research is to improve project duration and minimize unnecessary costs through the utilization of multi-skilled labor. Through a multi-step methodology, a model that optimizes the allocation of multi-skilled labor resources was developed. The novelty of the presented model is that it further minimizes the idle times of labor when compared to previous multi-skilled labor models, due to its capability in allocating resources "partially" to segments of activities rather than to full activities. In other words, unlike previous models, the developed model recognizes the fact that a crew can work for a period of time in an activity, then some workers in that crew can be allocated to another activity, leaving the rest of the crew to complete the first activity. The model allows the user to enter any number of activities and up to ten different resource types. With the use of genetic algorithms idle resources are assigned to activities that require additional manpower in order to reduce their durations, and in turn reduce

the project's indirect costs. When applied to a case study, the model generated promising results, where the reduction in duration between the single skilled allocation and multi-skilled labor allocation was 31% and this reduction jumped to 44% when partial allocation was applied. Multiskilling did not only reduce the idle labor days, but it will also shift the resource usage histogram's end point to the left, reducing the total project duration. This did not only reduce the unnecessary costs being paid to workers on days where they have no work, but it also reduced the total indirect costs which are directly proportional to the overall project duration.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1 - Introduction

Retaining and maintaining a skilled workforce on construction sites has always been one of the commonly faced problems associated with construction projects (Wang, et al., 2009). This is due to the fact that even during idle days, resources will still be assigned to the project since continuous hiring and letting go of workers could lead to the difficulty of maintaining top-quality workers due to unstable employment. This leads to unproductive levels of workforce on site that would keep some workers idle during periods of time where their demand is low. Many researchers have investigated different approaches to utilize workforce while increasing cost efficiency through the adaptation and allocation of multi-skilled resource allocation. On the other hand, most of the existing techniques for resource scheduling only consider single-skilled resource allocation. Even though single-skilled labor allocation is commonly used, it largely contributes to several factors leading to inefficient resource utilization in construction (Hegazy, et al., 2000).

1.1 Resource Allocation

1.1.1 Single-skilled Resourcing (Traditional Resource Allocation)

Resource allocation and scheduling has been widely studied and implemented in the construction industry. Researchers who examined the resource allocation problem were mostly tackling only single-skilled labor allocation. Single-skilled labor allocation utilizes the assignment of resource where each resource has the skill of only one trade. This however can result in several idle days where project managers may keep the idle workforce at their projects because they will be needed at a later stage and pay them in their idle days, which adds unnecessary costs to the project. Traditional resource allocation was tackled with the use of software programs such as Primavera and Microsoft Project. More advanced methods were also developed to solve the resource allocation problem. Such methods included but were not limited to Nassar (2005), who used genetic algorithms, an evolutionary algorithm technique, to develop a model that was designed to assign resources to a project with repetitive construction activities. Here the resource crew size was the main parameter used to optimize the project duration. El-Gafy (2006), on the other hand, used a different evolutionary algorithm technique, which utilized the ant-colony method for the allocation of resources in construction projects with activity repetition, ensuring that each of the workers had only one skill, to guarantee that each resource does not surpass its maximum resource availability. An alternative model that was developed by Moselhi and Alshibani (2007) that optimized resource allocation by integrating genetic algorithms with special technologies to determine the optimal crew assignments during the progress of site works, while accounting for the maximum available number of resources. Al-Bazi and Dawood (2010) presented a different model using genetic algorithm-based simulation modeling. The model's aim was to allocate resources in the precast concrete industry, which was then further developed by Al-Bazi and Dawood (2017) to optimize the resource costs, while of course taking into consideration constraints such as the available resource limit, skills of resources, crew formations and nature of the parallel repetitive layout of the manufacturing operations attributable to precast concrete. A schedule model for repetitive construction was proposed by Bhoyar and Parbat (2014) where precedence relationships were considered when assigning multiple crews, in addition to accounting for crew availability all while working on minimizing the project's overall time for completion and maximizing the continuity of resources on site. Francis Siu et al. (2015) recommended the utilization of a crew–job allocation model to facilitate resource management for both project and workface levels.

Alternative models considered the various qualities and characteristics that laborers may have. The characteristics of workers and how they interact to different site conditions is imperative when being measured in the allocation and scheduling analysis in construction to ensure the ideal deployment of each crew member.

1.1.2 Multi-skilled Resourcing

To optimize resources and account for practical considerations and limitations, resource leveling, or resource allocation would need to be applied; however, they mainly consider a single-skilled labor strategy. Single-skilled labor assumes that all the laborers on site are only capable of completing any tasks requiring the knowledge and expertise of only one skill. Some research on the other hand discussed the idea of "multi-skill" labor, where some of the workers may have enough training to carry out different activity types. If implemented correctly, it would reduce the project duration, since it would allocate idle laborers to other ongoing activities, which will increase the productivity for that specific activity and in turn decrease the overall activity duration (Cross, 1986).

Another problem would be determining where the worker/resource would be assigned if their skill is not required for another proceeding activity requiring that same skill. This could lead to the workers being hired, laid off, and then rehired again. However, when project managers assess whether it is better to keep workers on site during demand gaps or to lay them off, they mostly decide to keep them on site. This leads to the idle workers remaining on site, even though they are not assigned any tasks, in turn increasing the indirect costs on site since they will be paid their daily wages merely for showing up to work. It was found that the utilization of multi-skilled labor allocation has also proved to benefit the laborers as well as the project. Documented benefits include, lower turnover, and increased worker satisfaction as well as increased earnings for the workers (Alster, 1989).

1.1.3 Partial Allocation

Conducted research has only considered the allocation of resources to activities only for the entire activity duration, meaning that the resource must be assigned to the whole activity without the allowance of any preemption. Partial allocation on the other hand could allocate a resource to only part of an activity’s duration (Abotaleb, et al., 2014). Partial allocation was seldomly referred to in previous literature, however Abotaleb et al. (2014) suggested that the application of partial resource allocation would further benefit the optimization of resource allocation by decreasing activity duration, enhancing the continuity of resources on site and in turn improve cost efficiency. Figure 1 below demonstrates the difference between partial and non-partial resource allocation. The figure represents three activities; A, B and C, where resource R1 and R2 have been assigned to the whole durations of activities A and C. Given that R1 and R2 are available to be allocated to activity B between days 4 and 6, in non-partial allocation of R1 and R2 to activity B is not possible for two reasons: (1) the resources are already assigned to activity A during the first 2 days of activity B and (2) the resources are already assigned to activity C during the last two days of activity B. If partial allocation was permitted on the other hand, then resources R1 and R2 would be allowed to be allocated to activity B for two days, increasing the total resource productivity enough to reduce the duration of activity B by 33%. Therefore, the utilization of partial allocation in construction projects would yield very promising results in significantly reducing project durations, as will be further discussed in the upcoming sections.

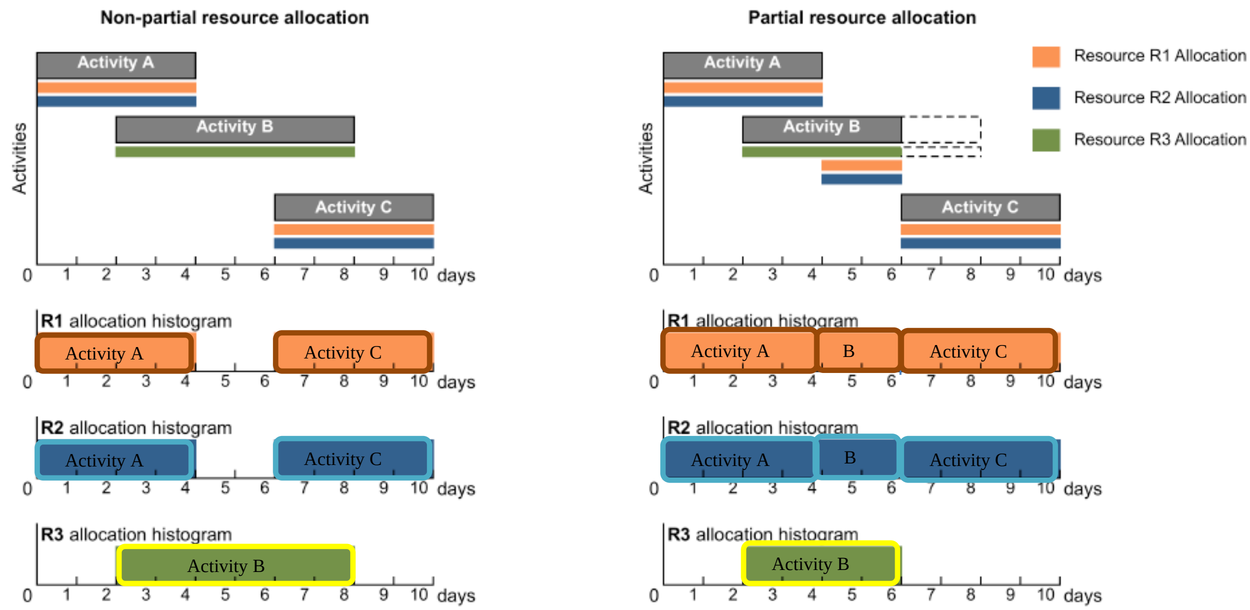


Figure 1: Concept of Partial Resource Allocation (Abotaleb, et al., 2014)

1.2 Previous Research on Multi-skilled Labor

Several papers and research were targeted on developing several scheduling techniques in order to find the optimum solution for resource leveling and allocation in order to find the solution able to result in the lowest indirect costs. Although the conducted research has achieved great results when applied to multiple construction projects, they mostly only took into account single-skilled labor as was demonstrated by Hegazy et al. (2000), where a framework was created in order to reduce indirect costs on site using the Early Late Start method heuristic approach to reduce the project durations by optimizing the allocation of resources. The Early Late Start method allocates available resources activities based on their late start values. Resources are assigned to the activities with the earlier late start values, until resource have been allocated to all activities.

Multi-skilled resourcing has been present in literature as early as the end of the 20th century, however research on optimizing multi-skilled labor allocation in the construction field was lacking. Nallikari (1995) employed a “multi-skill work team” techniques on shipbuilding facilities in Finland, which led to doubling the overall labor productivity. Another study was conducted by Carly in 1999 where approximately 1000 laborers in different locations and with different profession were asked to fill out a survey regarding the usage of multi-skilling. The results made it clear that multi-skilling would be preferred by the workers especially that it gave the workers stability knowing that they would stay on the project for longer and will learn new skills. Additionally, learning new skills would reimburse them through additional wages and benefits.

Gomar et al. (2002) described multi-skilled labor as laborers who have the ability and skill to participate in two or more different activities on site. This meaning that instead of only have the skills of a steel fixer, for example, the laborer would also be able to carry out plumbing tasks. If implemented correctly, this would reduce the project durations, since it would allocate idle laborers to other ongoing activities, which will increase the productivity for that specific activity, in turn decreasing the overall duration of the activity (Cross, 1986).

Wongwai and Malaikrisanachalee (2010) developed a model to reduce the total project cost and duration using multi-skilled resourcing. This model was based on fitness values assigned to determine the priority of rule-based parallel schedule generation on a local level and then proceeds to a more global solution. Even though this model did not use optimization techniques, it took into consideration additional overtime wages and the number of skills per worker were identified. The gap in this research was that it did not compare the output results with those with a model that could have been developed to show the costs and duration if a single-skilled labor model were used.

1.3 Problem Statement

Traditional allocation of resources in construction projects usually leads to unnecessary costs being paid just by keeping idle workers hired and paying their daily wage to keep them on site. Figure 2 below shows a traditional histogram of a labor resource, illustrating the frequency of idle days. The allocation of multi-skilled labor has been tackled previously, but the lack of utilizing partial allocation could restrict allocation of free resources that would lead to reductions in project durations if it were implemented correctly. Modeling multi-skilled labor with partial allocations would produce reliable and cost-effective results that could be implemented in real life projects to reduce both durations and boost expenditure efficiency.

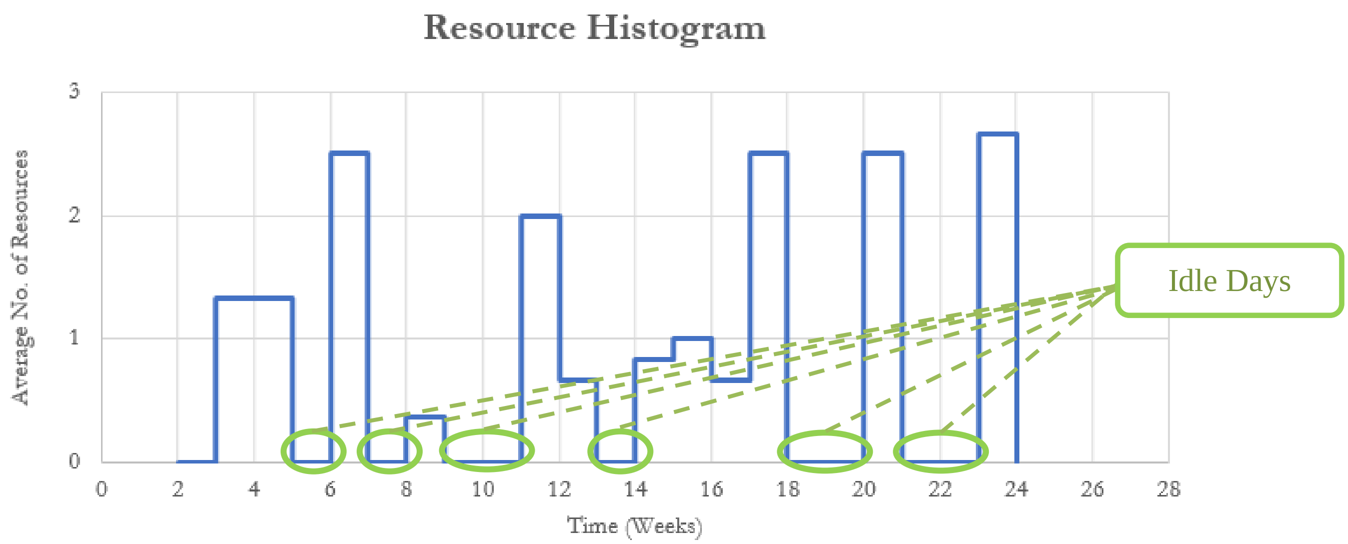


Figure 2: Resource Histogram Example Showing Working and Idle Days

The existing gap in literature revolves around the lack of extensive research when it comes to partial allocation in multi-skilled resourcing in construction projects. Previous research has also not shown much validation or verification of whether multi-skilled labor is more time and/or cost efficient than the traditional approach of single-skilled labor.

1.4 Research Objective

This research comprises of one goal, and several objectives as is listed below:

Research Goal:

- Improve project duration and minimize unnecessary costs through the utilization of multi-skilled labor.

Research Objectives:

- Develop a model for optimal allocation of multi-skilled labor with an emphasis on partial allocation of resources
- Study the impacts of partial allocation of multi-skilled labor on improving duration and cost.

The model would use the information entered by the user to create a framework that could accurately identify the optimum allocation of idle labor using genetic algorithms, taking into account the different labor costs and productivities entered by the user. This research will further expand on the previously conducted studies by introducing partial resource allocation. Partial allocation makes it more efficient to allocate resources during segments of activities - only during idle period of the resource - and then allows the secondary resource to go back to its primary activity. The model will calculate the labor costs if using multi-skilled skilled labor with and without partial allocation and will compare it with the same data but when using single-skilled labor.

It is also worth mentioning that multiskilling with not only reduce the idle labor days, but it will also shift the resource usage histogram's end point to the left, reducing the total time of each resource on site and hence reducing the total project duration. In addition to reducing the number of days idle labors are being paid, it will also reduce the total indirect costs which are directly proportional to the overall project duration.

1.5 Research Methodology

To meet the objective of this research, the research methodology is comprised of the following steps:

Part 1: Initial Investigation and Problem Identification

1- Review the literature to identify the gaps in research regarding multi-skilled resource allocation
2- Set the research objectives to solve the identified problem statement

Part 2: Model Development

1- Develop three different models for labor allocations, a) single skilled, b) multi-skilled without partial allocation and c) multi-skilled with partial allocation.
2- Integrate training costs with the multi-skilled resource allocation models
3- Determine the optimization technique to be used
4- Set the optimization procedure for the three models, by identifying the objective function, variables, and constraints.

Part 3: Verification, Validation and Analysis

1- Verify the developed models using ~~standard~~ verification tests
2- Validate the developed models using a case study,
3- Compare the outputs of each model to determine the efficiency of multi-skilling and partial resource allocation.

1.6 Thesis Organization

This thesis comprises of five main chapters listed below and demonstrated through figure 3:

Chapter 1 – Introduction:

This chapter provides an introduction on resource allocation, and the advantages of applying multi-skilled resource allocation, in addition to identifying the problem statement, research objectives and methodology, and finally the thesis organization.

Chapter 2 – Literature Review:

In this chapter, an all-encompassing analysis of the previous research regarding multi-skilled labor was conducted.

Chapter 3 – Model Development and Formulation:

Here, the model development is broken down to in-depth details, to demonstrate how to model is used to satisfy the research objectives of developing a multi-skilled labor allocation model with partial allocation, which has never been conducted before.

Chapter 4 – Case Study and Model Validation:

This chapter shows the application of the model through a case study, which represents a real construction project. The results are compared to validate the model.

Chapter 5 – Conclusion and Recommendations:

The research is summarized and any recommendations for future research and improvement in multi-skilled resource allocation are discussed.

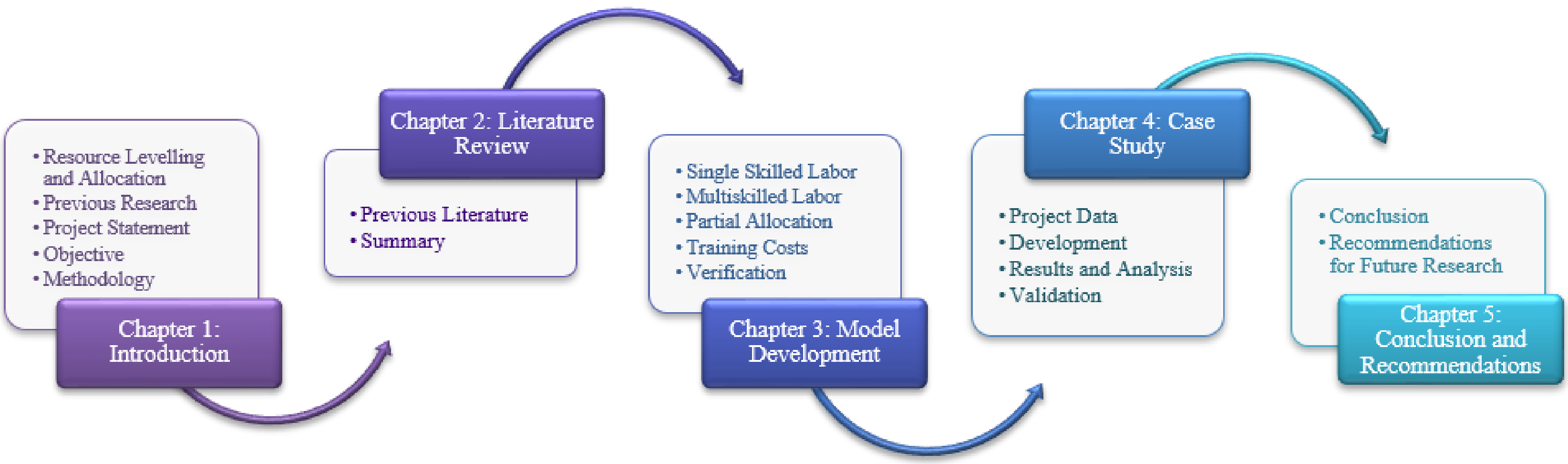


Figure 3: Thesis Organization

Chapter 2 - Literature Review

2.1 Multiskilling Categorization

Different techniques have been identified and used in previous research to tackle the multi-skilled resource allocation problem. The previous conducted research related to multiskilling was found to have experienced academic, industrial in addition to governmental points of views. This chapter aims to cover an extensive analysis of the current literature, covering the classifications of the effects of multiskilling to present the concept in a clearer manner, and to identify any gaps in the existing research. The papers discussed and presented in this chapter include papers from the initial mentioning of the concept of multiskilling to the most recent papers where multi-skilled resource allocation models have been developed and tested.

The concept of multiskilling goes back to the 1970's where it was discussed to utilize workforce organization in manufacturing plants. Bittel & Trepo (1979) and Liu & Ortsman (1979) both recommended the application of multiskilling in a new production plant and a chemical plant, respectively. The conducted research focused on the social impacts affecting the workers and was intended to increase worker qualifications and salaries.

The concept of multi-skilled labor was first introduced to the construction industry domain by Burleson et al. (1998). Multi-skilled labor was defined by Gomar et al. (2002) as laborers who have the ability and skill to participate in two or more different activities on site. It was found that some studies had retrieved their techniques in tackling multiskilling in construction project from previous research conducted in the service/manufacturing industries (Arashpour, et al., 2017). However, there are significant differences between the manufacturing and construction industries.

Other studies targeted their focus on dual resource constrained systems, where the selected resources could learn a maximum of only two different skills, i.e., one additional skill to the main skill that the worker is already proficient at (Treleven, 1989; Hottenstein & Bowman, 1998). It was found that there is a wide range of existing methods and conducted research on the implementation of multiskilling with several tools such as dynamic scheduling and work study to enhance productivity rates (Pagano & Heathcote, 2003).

Multiskilling is a strategy used to manage construction workforce to increase flexibility in the allocation of resources by enhancing a certain skill set to broaden out and include additional skill sets to be assigned whenever needed (Haas, et al., 2001). Multitasking has proved to play a noticeable part in the improvement of construction performance (Ahmadian Fard Fini, et al., 2017).

Even though there are several different adopted strategies and techniques that were developed when it comes to multi-skilled labor scheduling, they more or less only target a fixed set of objectives which include, but are not restricted to: improving prospects of labor employment (Haas, et al., 2001; Wang, et al., 2009), reducing of labor costs to a minimum (Gomar, et al., 2002; Srour, et al., 2006), overcoming workforce shortages (Pollitt, 2010), improving of safety (MOM, 2016), and finally boosting labor productivity (Florez, 2017).

Benefits:

According to Tatum (1989), Nonaka (1990) and Ettlie & Rezo (1992) there are several benefits to multi-skilled resource allocation which include but are not limited to the following:

- Considerable reductions in project costs, primarily due to the savings of project labor costs
- Reduction in the number of workers assigned to the project
- Implied benefit of increase in annual employment rates
- Average durations of employment increased which results in higher job stability for the workers

Limitations:

Although multitasking can indeed bring several benefits to both the employer and the workers, the implementation of multi-skilled resource scheduling also has limitations and collateral effects. For example, a worker with a certain skill may be unwilling to undergo additional training for several reasons, with those reasons including not accepting to the idea of learning another skill that the worker would regard to be unsuitable of his skills (Ho, 2016), or that the worker believe that they already have sufficient knowledge to carry out the task requiring the other skills, or that the worker would worry that they might not receive enough compensation in exchange for the additional training (Carley, et al., 2003). Other limitations that would most likely be faced include decreased efficiency in productivity when carrying out a task that required the newly attained skills (Wang, et al., 2009). The main limitation, however, was found to be insufficient funding to cover the expenses of training costs (BCA, 2016).

2.2 Problem-Solving Techniques for Multi-Skilled Labor Allocation

Several methods have been developed to solve the resource allocation problem, including multi-skilled resourcing. These techniques include mathematical heuristic methods, linear programming, mixed-integer linear programming, and evolutionary algorithms. Several papers and research were targeted on developing several scheduling techniques to find the optimum solution for multi-skilled resource leveling and allocation to find the solution able to result in the lowest indirect costs. According to conducted research, the most widely used optimization techniques that could be implemented are genetic algorithms and heuristic techniques (Project Management Institute , 2017), which are further elaborated on in the upcoming sub-sections.

2.2.1 Heuristic Methods

Heuristics are problem solving experience-based methods that provide solutions in limited time frames based on a series of consecutive steps. In other words, heuristics reduce the need to do complex calculations for large problems through segmenting the solving methodology into a series of steps to reach the final solution. The solutions obtained by heuristics are not necessarily optimal, however they provide sufficient and satisfactory solutions (Pearl, 1984). Although multiple research papers have tackled the problem of resource-constrained scheduling, Hegazy et al. (2000) and Wongwai & Malaikrisanachalee (2010) were of the few who developed heuristic models that solved the multi-skilled allocation alternative. The heuristic approach demonstrated by Hegazy et al. (2000), essentially modified the single-skill resource allocation technique to create a framework directed at reducing indirect costs on site using the Early Late Start method heuristic approach to reduce the project durations by optimizing the allocation of resources. There were resource substitution rules that needed to be satisfied for the model to work. For example, resource allocation was only applied for activities with higher priorities (early late start dates), irrespective of the any concurrent activity’s resource requirements. Here the model does not allow the allocation of any resources to an activity except if resources necessary for the completion of that activity can be entirely met. All available resources in the resource pool are only directed towards the activity with the lowest early start, and in turn any other activities cannot start unless it becomes the activity with the highest priority and its required resource is available. The model was tested on a case study that proved its powerful capabilities as the overall project duration was reduced from 49 days with single skilled resource constraints to 39 days when multi-skilled resource constraints were applied, giving a 20% reduction in duration.

Wongwai & Malaikrisanachalee (2010) approached multi-skilled resourcing from another perspective, overcoming the “all-or-nothing resource assignment concept” improving the resource substitution technique. Here, resource allocation is also implemented using the Earliest Late Start method, where resources are assigned vertically. Each of the current eligible activities are assigned with their resource requirements, depending on priority order irrespective of any resource limitations. An activity will only be delayed if the number of qualified resources is not sufficient to satisfy the required resources needed for an activity. In this case the activity is delayed until the nearest day where more resources that were allocated to other concurrent activities become available. Any remaining resources not required by the delayed activity are then released back to the resource pool to be allocated to the next eligible activity, until enough resources are available to fulfil the requirements of the previously delayed activity. The model formulation is demonstrated further through figure 4.

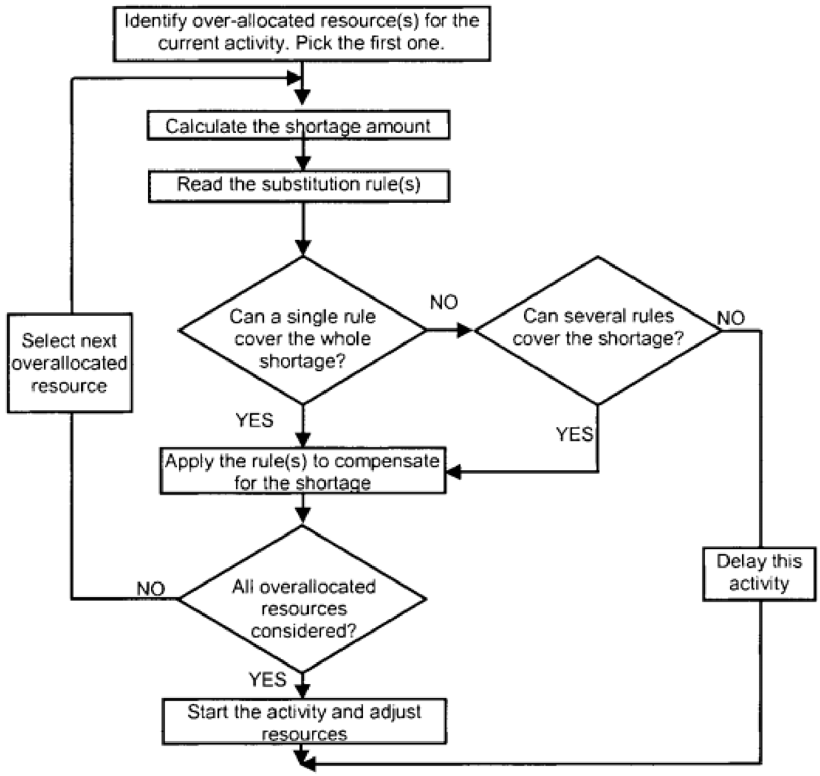


Figure 4: Model Formulation (Wongwai & Malaikrisanachalee, 2010)

The concept that was developed to determine the progress work from the proportional between assigned and required labor were as follows:

$E_{(i,j,r)} = \frac{a_{(i,j,r)}}{q_{(i,r)}\ x\ d_{(i)}}$ (Wongwai & Malaikrisanachalee, 2010)

$P_{(i,j)} = \min\{E_{(i,j,r)} | r = 1,2, \dots, N_{i,j}\}$ (Wongwai & Malaikrisanachalee, 2010)

$C_{(i)} = \min\{100\%, \sum_{j=S_{(i)}}^{c_i} P_{(i,j)}\}$ (Wongwai & Malaikrisanachalee, 2010)

Where:

$E_{(i,j,r)}$ = earned progress of activity *i* on day *j* from resource *r*

$a_{(i,j,r)}$ = number of assigned resources *r* to activity *i* on day *j*

$q_{(i,r)}$ = daily requirement of resource *r* for activity *i*

$d_{(i)}$ = original duration of activity *i*

$P_{(i,j)}$ = progress of activity *i* on day *j*

$N_{i,j}$ = the last resource that is assigned to activity *i* on day *j*

$C_{(i)}$ = the cumulative progress of activity *i*

$S_{(i)}$ = start date of activity i

c_i = current date of activity i

Furthermore, in order to improve the final results in terms of project duration, two additional steps were applied before starting resource substitution, with the first step being assigning resources to all of the activities that are eligible to start, after which qualified resources would be taken back from a lower priority if the remaining qualified resources are insufficient. The second step would be constraining the start of each activity with a certain number of resources. An activity that is not team constrained would start regardless of the insufficiency of resources. This increases the starting opportunities of activities at earlier times, which did indeed achieve better results than the previously existent heuristic approach.

2.2.2 Mathematical Techniques

Mathematical optimization is defined as the selection or determination of the best solution from a set of alternatives or variables, while considering some constraints or criteria. An optimization problem can simply be described as maximizing or minimizing a set value by choosing the best possible value from variables through conducting multiple different iterations. Due to the ability of mathematical techniques to solve complex engineering problems, several multi-skilled labor allocation models were developed using mathematical techniques such as linear programming, evolutionary algorithms and integer programming.

2.1.2.1 Linear Programming

One of the major subfields of mathematical techniques is linear programming, which is a type of convex programming. In convex programming the objective function is defined by either minimizing or maximizing a certain parameter. Linear programming was developed in by George Dantzig in 1947, who discovered that there was generalization in the mathematics of scheduling and planning problems (Pike, 1986). Here, the relationships are represented in a linear manner.

Some of the conducted research tackled the multi-skilled labor allocation problem using linear programming. Gomar et al. (2002) recognized the need of cross-skilled labor to overcome the drop in labor availability in the market with multiple objectives; minimizing hires, maximizing usage of each resource throughout project duration, and minimizing resource reallocation. Gomar et al. (2002) described multi-skilled labor as laborers who have the ability and skill to participate in two or more different activities on site. This meaning that instead of only have the skills of a steel fixer, for example, the laborer would also be able to carry out plumbing tasks. The aim of that research was to test the multi-skilled optimization problem and validate its results using the Construction Industry Institute Model Plant data and a commercial linear programming software. The Model plant was developed by the CII, and its role is to provide a standard reference to physical productivity measurements. One of the studies conducted using the Model Plant was in the development of an economic model for a multi-skilled workforce. The model formulation was based on the model formulated in an earlier report by Gomar (1999). Briefly speaking, the model allocates resources from a predefined resource pool in the database. The model has the capability of determining which resources to hire when, and when to lay off a resource completely if it has completed all the activities that it could be assigned to. The model's mathematical formulation is extensively presented through figure 5, with the objective function being determined through three factors: (1) to minimize the total number of workers, (2) minimize switching of resources between activities, and (3) minimizing hiring and firing of resources, and in turn minimizing the project cost. The model's results showed that it could optimize the allocation of a partially multi-skilled workforce and that the project gains the benefit of multiskilling if only 10-20% of multiskilling is applied, anything after that is marginal.

Srour et al. (2006) also developed a linear programming model and verified it using the CII model plant data, in which the model was able to provide a strategy for training and hiring workers to satisfy schedule requirements. The objective function was also to minimize the construction labor costs while ensuring that the labor demand profile is met over the duration of the project. This model improved on the model developed by Gomar et al. (2002) by considering that single skilled labor could still be trained on site, and in turn also considering training costs. The constraints were: (1) the training capacity of each skill, (2) meeting the demand required by each resource, (3) hiring limit, and finally (4) resource availability. The model's results showed that the benefit-to-cost analysis after the implementation of the model significantly increased. The model however, like several other previous models, did not try to tackle the reduction of project durations through the allocation of multi-skilled labor. The model was applied to a case study which showed that the model saved around $32,000 in labor costs (after deducting all training costs) from approximately $9 million total labor costs. (Hyari, et al., 2010).

Sets:

I set of workers
J set of jobs
T time periods
I_j subset of workers that can do job j
J_i subset of jobs that worker i can do

Data:

$DEMAND_{jt}$ demand of skill j on day t
P objective penalty weight on switching
F objective penalty weight on hires & fires

Decision Variables:

W_{it} worker i hired or allocated on day t
X_{ijt} worker i assigned to do skill j on day t
Yp_{ijt} positive counter variable that measures switching of worker i using skill j on day t
Yn_{ijt} absolute value of negative counter variable that measures switching of worker i using skill j on day t
Z_{it} hires counter variable of worker i on day t

$$Minimize \quad \sum_{i\in I}\sum_{t\in T} W_{it} + P * \sum_{i\in I}\sum_{j\in J_i}\sum_{t\in T}(Yp_{ijt} + Yn_{ijt}) + F * \sum_{i\in I}\sum_{t\in T} Z_{it}$$

$s.t.$

$$\sum_{i\in I_j} X_{ijt} = DEMAND_{jt} \ldots\ldots\ldots\ldots \quad \forall\, j\in J, t\in T$$

$$\sum_{j\in J_i} X_{ijt} \le 1 \ldots\ldots\ldots\ldots \quad \forall\, i\in I, t\in T$$

$$X_{ijt} \le W_{it} \ldots\ldots\ldots\ldots \quad \forall\, i\in I, j\in Ji, t\in T_i$$

$$Yp_{ijt} - Yn_{ijt} = X_{ijt} - X_{ij(t-1)} \ldots\ldots\ldots\ldots \quad \forall\, i\in I, j\in J_i, t\in T, t\neq 1$$

$$Z_{it} \ge W_{it} - W_{i(t-1)} \ldots\ldots\ldots\ldots \quad \forall\, i\in I, j\in J_i, t\in T, t\neq 1$$

where W_{it}, X_{ijt}, Yp_{ijt}, Yn_{ijt}, Z_{it} $\in \{0,1\}$ $\forall\, i\in I, j\in J_i, t\in T$

Figure 5: Mathematical Formulation of the Model Developed by Gomar et al. (2002)

2.1.2.2 Mixed Integer Linear Programming

Integer programming is also a mathematical formulation which is very similar to linear programming, with the only difference being constraining the variables to being integers only. Mixed Integer Linear Programming on the other hand is a hybrid between both, where some of the variables must be integers only, while other variables can be non-integers.

This technique was used by Hyari et al. (2010) in presenting a different approach to tackling the multi-skilled labor problem. The objective was to minimize the cost of labor required to perform the project activities. The unit cost of performing an activity is a function of the hourly wage the worker resources and their productivity. The objective function was accomplished by minimizing the cost of labor using the equation below:

$$Minimize \sum_{i=1}^{I} \sum_{j=1}^{J} X_{ij} \; x \; \frac{W_j}{Pr_{ij}} \qquad \text{(Hyari, et al., 2010)}$$

Where:

X_{ij} = Quantity of activity *i* carried out by resource *j*

W_j = hourly wage of resource *j*

Pr_{ij} = productivity rate of resource *j* in carrying out activity *i*

Subsequently, the number of resource crews required by each activity was achieved by dividing the quantity assigned to each resource by the corresponding productivity of that resource. Model constraints were put into effect to ensure that the model is practical enough and can correspond to with actual multi-skill tasking on site. There were five constraints set in the model: (1) work assigned to the resource should be equal to the quantity of the activity, (2) the number of resources assigned per activity is in integers, (3) resources selected for a job should only be within the resources that can perform the activity, (4) all variables need to be greater than or equal to zero and (5) the number of crews assigned from each resource should not surpass the limit of available resources.

The model used mixed integer programming, which is an extension to the linear programming method. The variables were constrained to having only integer values at the obtained optimal solution, which was implemented using Excel Solver. The model’s results were compared with a previously developed model. The new model yielded better results since it produced results in a shorter period and the results were guaranteed to produce a globally optimal solution, and best of all the model attained a total project cost that was less than that reached by the other model.

2.2.3 Evolutionary Algorithms

Evolutionary algorithms are advanced heuristics techniques that require higher processing power and the use of computer intelligence to generate near-optimum solutions. There are several problems in the construction industry that require the use of computer intelligence to solve them. These computer intelligence techniques can be categorized into evolutionary algorithms, neural networks, and fuzzy logic. Throughout the several research efforts conducted regarding this aspect, it was found that many of the proposed methods using evolutionary algorithms opted for the genetic algorithms method in order to find the most suitable option in terms of allocating the right resources to the most suitable activities, as opposed to the limited resource leveling abilities of Primavera (Gomar, et al., 2002).

Tam et al. (2001) defined genetic algorithms as being a “heuristic random search technique based on the concepts of natural selection and natural genetics of a population; therefore they are a ‘population-based’ method of searching large combinatorial spaces to find the near optimal combination”. When the resource allocation problem is converted to strings, genetic algorithms can be utilized to support solving the problem. Genetic algorithms derive their power from the mechanics of natural selection and the survival-of-the-fittest principles (Goldberg, 1989). Genetic algorithms was used in the development of an optimization model that optimized the deployment of labor. The developed model was designed to minimize the labor costs by allocated resource crews to different skill types to carry out a set number of trades in a specified time limit.

Figure 6 presents the framework of the proposed model, which shows that the process is comprised of 15 steps. The first few steps define the inputs module which includes the following: the daily wages of each resource, the nature of the resource’s skill, the productivity rates of each resource type when carrying out a certain activity type, daily quantity requirements that need to be met and resource availability limits.

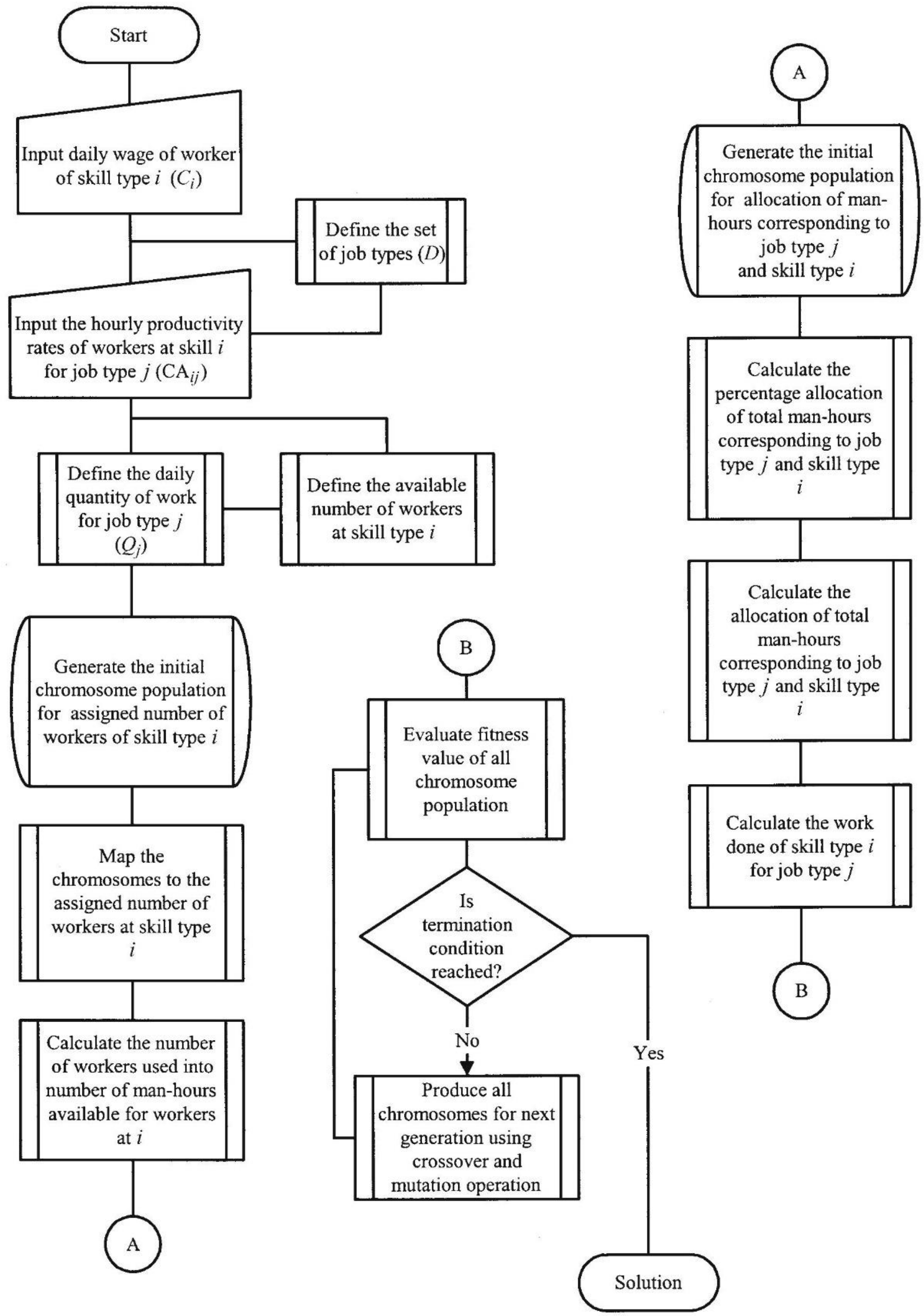


Figure 6: Flow Chart of the GA Model Proposed by (Tam, et al., 2001)

Once these inputs are defined, the data is then transferred to the optimization module, where the chromosomes for the number of workers required to complete a certain task are defined. With this step complete, the chromosomes are then mapped and transferred to calculate the man-hours. Other chromosomes are generated for the percentage allocation of man-hours per task. When these chromosomes are mapped, the number of man hours is calculated as well as the quantity of production for the workers. The sum of productivity of workers of all skill types was constrained to being greater than or equal to the quantity of work. The fitness function is then calculated and evaluated. For the crossover and mutation stage, a single point continuous crossover is selected as the crossover operator. Finally, a near optimal solution is reached when the termination condition is met.

The model was tested out using a case study where three scenarios were simulated and compared together, with the scenarios being: (1) the available resource surpasses the number of required resources, (2) some skill types are lacking and (3) the supply of work of a particular trade was increased, with the objective function being the minimization of the labor costs for all three simulations. The model showed effective results in attaining optimal allocation of resources, by utilizaing the use of the existing resource pool giving an optimal total labor cost in multi-skilled labor procedures.

Long & Ohsato (2009) also used the genetic algorithm technique to determine suitable start times of activities in repetitive construction. This model had three objectives: minimizing project duration, minimizing total project cost, in addition to minimizing the combined performance of the total project cost and project duration, as per the equation below. The overall project objective is calculated by assigning weights that signify the importance of project durations and total project cost.

$$\mathrm{TC} = \left[\mathrm{W_t}.\left(\frac{\mathrm{T_p} - \mathrm{T_p^{Min}}}{\mathrm{T_p^{Min}}}\right)^2 + \mathrm{W_c}.\left(\frac{\mathrm{C_p} - \mathrm{C_p^{Min}}}{\mathrm{C_p^{Min}}}\right)^2\right]^{1/2} \qquad \text{(Long \& Ohsato, 2009)}$$

TC represents the total combined performance of the project in terms of duration and cost. Wt represents the weight assigned to the duration factor and Wc is the weight assigned to the cost factor. The paper indicated that the weights were based on the user's preference. If both weights were 50%, this would mean that both cost and time have equal weights. The model is run a number of times, while setting different objectives each time. The first objective function was to find the minimum cost ($\mathrm{C_p^{Min}}$), and a second yet separate objective function was to find the minimum project duration (T_p^{Min}). With the minimum cost and minimum duration values determined, the model is run a third time to minimize TC.

Liu & Yang (2011) developed a model to reduce the total project cost and duration through the use of multi-skilled resourcing. This model was based on fitness values assigned to determine the priority of rule-based parallel schedule generation on a local level and then proceeds to a more global solution. Even though this model still done not work on optimization techniques, it took into consideration additional overtime wages and the number of skills per worker was identified. These inputs were not taken into consideration in previous models. The gap in this research was that it did not compare the output results with those of a model that could have been developed to show the costs and duration if a single-skilled labor model were used. The results showed that the model performs accurately and efficiently for small-sized problems and can obtain exact solutions for most cases.

Abotaleb et al. (2014) on the other hand, also developed a multi-skilled labor allocation model utilizing the deployment of labor using genetic algorithms but with the single objective function to minimize project duration. The model contained two constraints, with the first being setting a limitation on the available number of resources per day to ensure proper resource leveling and to avoid over-allocation in any activities. The second constraint was the number of different resource types that can be assigned to any one activity. This was defined as a soft constraint, and its only role was to avoid over-crowding. Furthermore, the research conducted compared the results from a single-skilled labor model with a multi-skilled labor model demonstrated the jump in cost efficiency and duration reduction after the deployment multi-skilled labor. When comparing the results of single-skilled with the multi-skilled labor allocation models, it was found that there was a 31% and 30% reduction in project duration and overall project cost respectively.

2.3 Summary of Literature Review

2.3.1 Optimization Techniques

Several different methods were used to tackle the allocation of multi-skilled labor to reduce project costs and/or duration. Heuristic techniques may not result in near optimum solutions, however mathematical methods, such as linear programming, mixed integer programming and genetic algorithms were able to recognize and solve the problem through setting the objective function (by either minimizing or maximizing it), depending on a certain set of variables and constrained by another set of boundaries.

2.3.2 Objective functions

Further to categorizing conducted research based on the technique used, it can also be categorized according to the objective function of the model. The three main objective functions that were targeted were (1) minimizing total project duration, (2) minimizing total project cost through the minimization of labor costs, and (3) social sustainability. Social sustainability takes into consideration the social impacts affecting the workers in terms of increased qualifications and salaries.

2.3.3 Research Gap

The existing gap in literature revolves around the lack of extensive research when it comes to partial allocation in multi-skilled resourcing in construction projects. Previous research has also not shown much validation or verification of whether multi-skilled labor is more time and/or cost efficient than the traditional approach of single-skilled labor.

CHAPTER 3 - MODEL FORMULATION AND DEVELOPMENT

This chapter details the formulation of the model structure in which the algorithm that optimized multi-skilled labor with and without partial allocation was developed. The model targets the utilization of multi-skilled labor to minimize the presence of idle labor on site. The model is intended to be generic and therefore would allow the user to input as many activities and as is required by the project. As the user enters the necessary inputs, the model would use genetic algorithms to allocate the idle resources to activities that require additional manpower in order to reduce their durations, in turn reducing the indirect costs impacted onto the project. The purpose of the model is to utilize all the resources in an efficient manner, since not only does it allocate resources for the entire duration of an activity, but it also allows for partial allocation, which will be further detailed in the upcoming sections.

3.1 Model Architecture

The point of this research is to develop a program which efficiently allocates idle labors to ongoing activities from other trades while ensuring that the project cost remains within the assigned budget. The allocation of idle resources during partial durations of activities also reduces the duration assigned to activities that were generated after applying the single-skilled labor algorithm. The developed model is mainly formed of four different modules. These modules comprise of the: (1) inputs module, (2) calculations module, (3) optimizations module and (4) outputs module. The user enters the project information in the first module, after which these inputs are then used by the calculations module, which is then optimized to generate the output module, as is demonstrated through figure 7. The concept behind the model is primarily based on the model developed by Abotaleb et al. (2014), however this research further builds on it by introducing and implementing multi-skilled labor to full and partial activity durations.

Traditionally, during the execution of any construction project, each activity is completed by one resource type throughout the duration of the activity. With multi-skilled labor, previous research was able to allocate the secondary resources only during the full durations of activities. With the introduction of partial allocation, resources can be allocated accurately according to their idle time only without the secondary activity duration being of any relevance. This increases the ability of the model to allocate resources to significantly reduce any, if not all, idle man-hours, and consequently reducing the project duration, while maintaining the project cost, which adds to the novelty of this model.

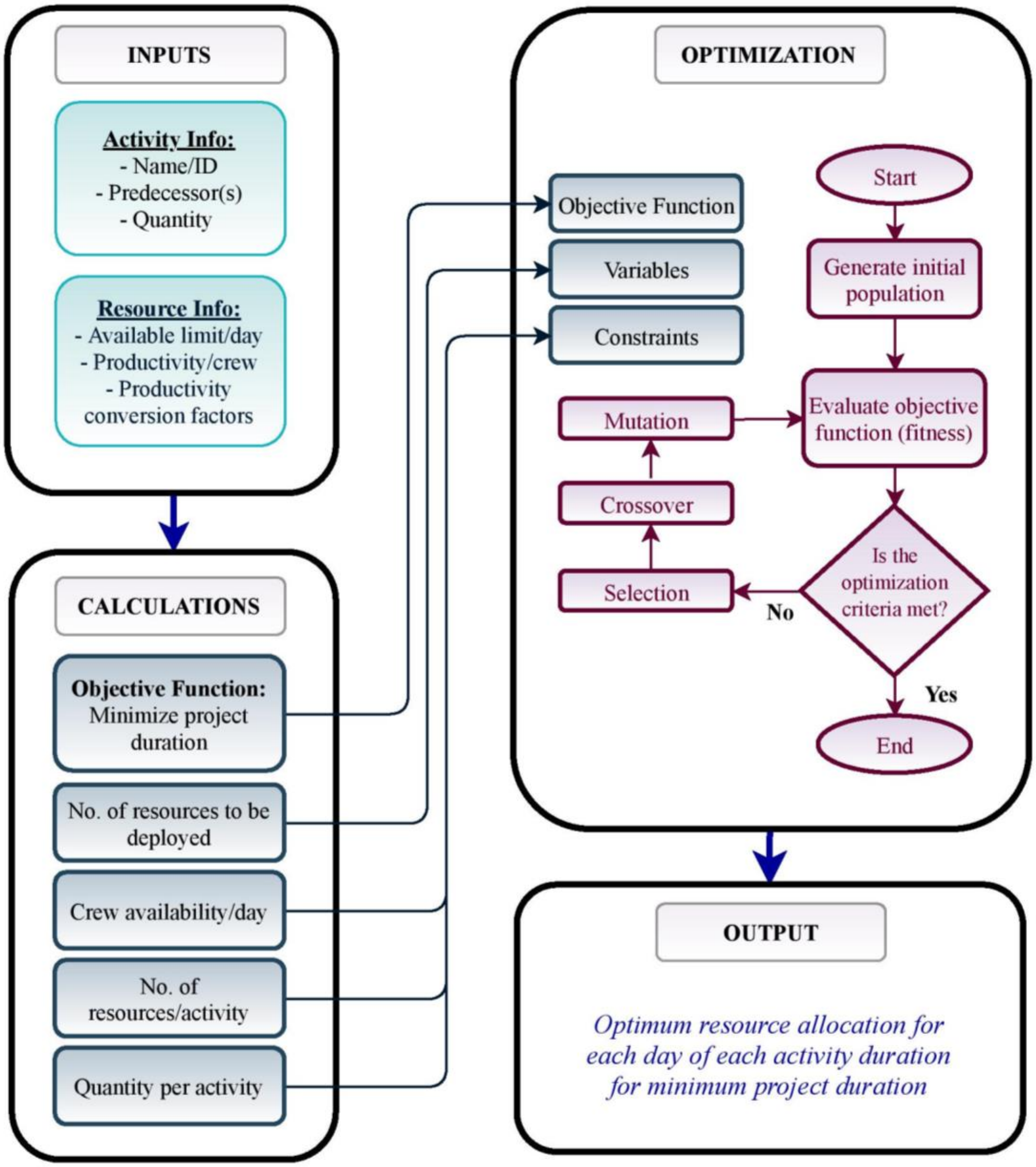


Figure 7: Model Development Methodology

3.1.1 Inputs Module

The model allows the user to enter the input data which comprises of the following data:

- Activity name
- Activity ID
- Activity predecessors
- Quantity of activity
- Primary resource
- Productivity of each primary resource
- Productivity conversion factors for secondary resourcing
- List of all resources needed for the project
- Maximum available crews for each resource
- Daily wage for each resource
- Training costs

The model reads and uses the input data to optimize the allocation of multi-skilled labor. The developed software is a user-friendly model, which allows the user to enter an infinite number of activities, having any number of predecessors with finish-to-start relationships. The model was developed using Microsoft Excel and Evolver which is a genetic algorithms plug-in. To allow for user-interface, the model was integrated with Visual Basic Algorithms. Visual Basic is built in Microsoft Excel to help write programs for the windows operating system through coding. To enable the user to enter all the project information and inputs, Userforms are created. A Userform is simply a window, which interacts with the user by showing a set of instructions to guide the user on how to enter the required information. For this model, the user is directed to the first Userform, which is a quick welcome message, giving a quick overview of what the software does, after which the user will be required click on a “next” button, which once again directs the user to the following window. For this model, the following userform (shown in figure 8) appears for the user to enter each activity along with its predecessor(s), quantity and primary resource required to complete the activity and all the required input information. Five sets of Userforms were created, one for activity related information, and the other for resource related information. The Userforms integrated with the model are shown through the figures below.

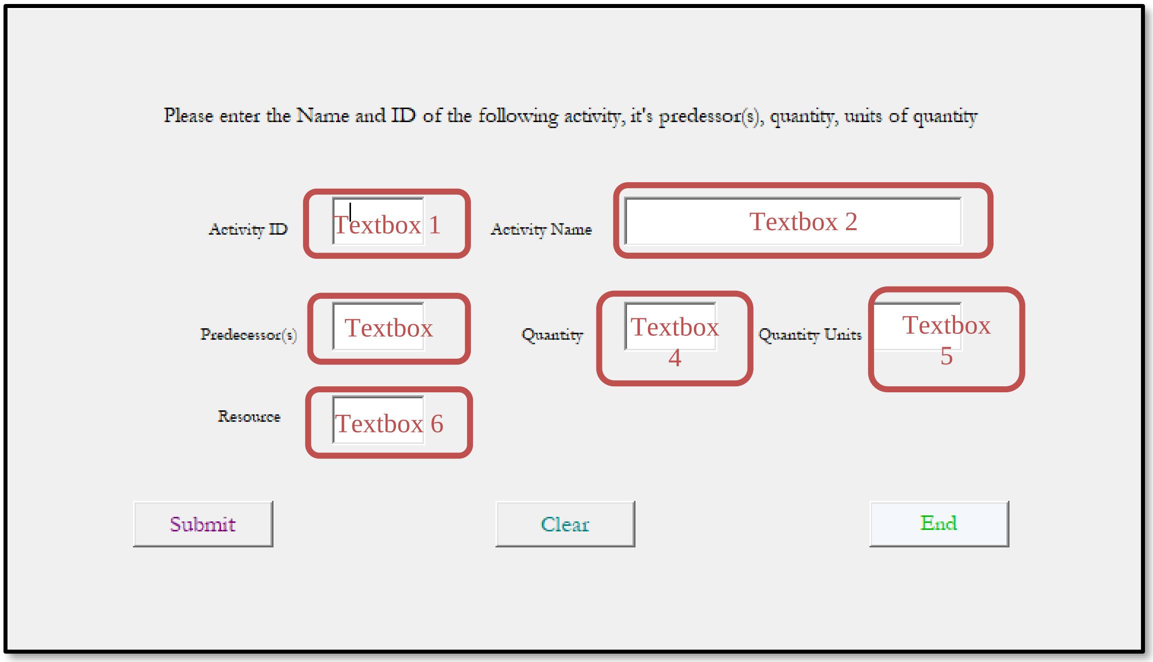


Figure 8: Userform 2 – Activity Inputs

The first userform is a welcome message introducing the user to the program. Userform 2 allows the user to enter the first set of inputs; the activity related inputs. Usually, each userform can be created to receive one set of data; however, to allow the user to keep entering data until all the activities have been entered, a code was developed to allow for a loop of the same userform to come up until the user decides that all the information has been entered, after which the inputs module is finalized. These inputs include activity name and ID, in addition to quantity, predecessors and the primary resource of each activity. The code shown below (figure 9) is linked to the “Submit” button shown in figure 8. This code enables the same userform to appear on repeat for the user to enter the information of each activity into the user inputs worksheet as is shown in figure 10. Every time the user clicks on “submit” the entered information is entered into the consequent row with an offset of one row at a time.

```
CommandButton4

Private Sub CommandButton2_Click()
Dim wksl As Worksheet
Dim AddNewl As Range

Set wksl = Sheets("UserIn")

Set AddNewl = wksl.Range("A65356").End(xlUp).Offset(1, 0)

AddNewl.Offset(0, 0).Value = TextBox1
AddNewl.Offset(0, 1).Value = TextBox2
AddNewl.Offset(0, 2).Value = TextBox3
AddNewl.Offset(0, 3).Value = TextBox6
AddNewl.Offset(0, 5).Value = TextBox4
AddNewl.Offset(0, 6).Value = TextBox5

End Sub
```

Figure 9: Code Used to Transfer Activity Inputs from Userform to Worksheet

	A	B	C	D	E	F	G
3 4	Activity ID	Activity Name	Predeccessors	Required Resources	Resource Code	Quantity	Units
5–10	(0,0)	(0,1)	(0,2)		(0,3)	(0,5)	(0,6)

Figure 10:Sample of User Inputs Worksheet Receiving Activity Information

Once the user enters all the project activities, the user can then click on “end”, moving on to the next userform. Userform 3 (figure 11) is to collect resource related information, such as productivity, daily wage, and maximum available limit per day in addition to assigning each resource to a set resource code. The entered data is transferred to the user inputs worksheet shown in figure 12, and the transfer is achieved using the code in figure 13.

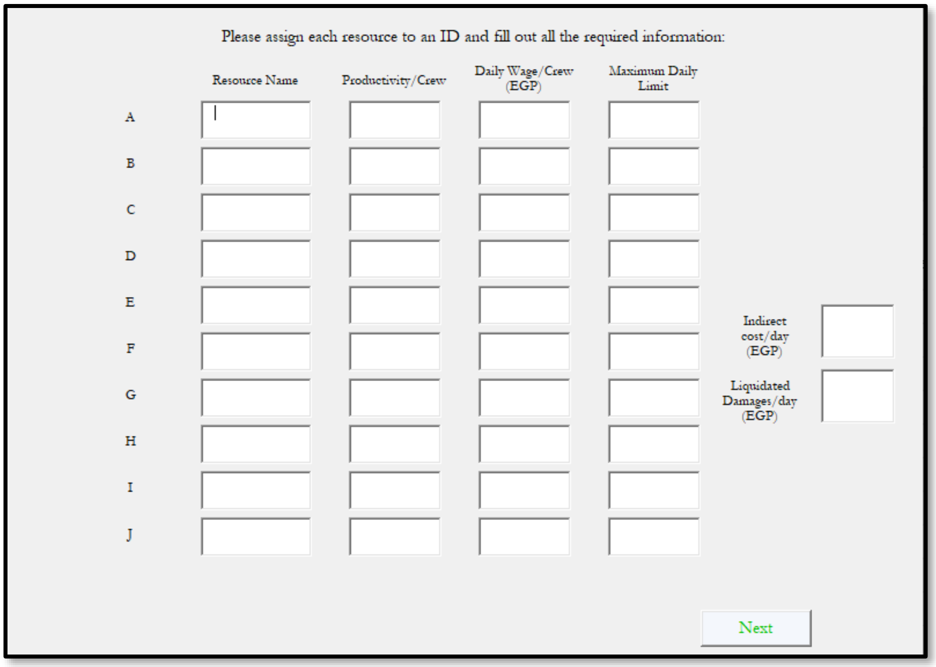


Figure 11: Userform 3 - Resource Inputs

	I	J	K	L	M
3–4	Name	Resource Number	Productivity	Daily Wage	Max No. of Resources
5		A			
6		B			
7		C			
8		D			
9		E			
10		F			
11		G			
12		H			
13		I			
14		J			

Figure 12: Sample of User Inputs Worksheet Receiving Resource Information

CommandButton4 Click

```
Private Sub CommandButton4_Click()
Unload UserForm3
Sheets("UserIn").Range("I5").Value = TextBox1.Value
Sheets("UserIn").Range("I6").Value = TextBox3.Value
Sheets("UserIn").Range("I7").Value = TextBox6.Value
Sheets("UserIn").Range("I8").Value = TextBox7.Value
Sheets("UserIn").Range("I9").Value = TextBox9.Value
Sheets("UserIn").Range("I10").Value = TextBox10.Value
Sheets("UserIn").Range("I11").Value = TextBox8.Value
Sheets("UserIn").Range("I12").Value = TextBox12.Value
Sheets("UserIn").Range("I13").Value = TextBox13.Value
Sheets("UserIn").Range("I14").Value = TextBox11.Value

Sheets("UserIn").Range("K5").Value = TextBox15.Value
Sheets("UserIn").Range("K6").Value = TextBox16.Value
Sheets("UserIn").Range("K7").Value = TextBox17.Value
Sheets("UserIn").Range("K8").Value = TextBox18.Value
Sheets("UserIn").Range("K9").Value = TextBox20.Value
Sheets("UserIn").Range("K10").Value = TextBox21.Value
Sheets("UserIn").Range("K11").Value = TextBox19.Value
Sheets("UserIn").Range("K12").Value = TextBox22.Value
Sheets("UserIn").Range("K13").Value = TextBox23.Value
Sheets("UserIn").Range("K14").Value = TextBox14.Value

Sheets("UserIn").Range("L5").Value = TextBox25.Value
Sheets("UserIn").Range("L6").Value = TextBox26.Value
Sheets("UserIn").Range("L7").Value = TextBox27.Value
Sheets("UserIn").Range("L8").Value = TextBox28.Value
Sheets("UserIn").Range("L9").Value = TextBox30.Value
Sheets("UserIn").Range("L10").Value = TextBox31.Value
Sheets("UserIn").Range("L11").Value = TextBox29.Value
Sheets("UserIn").Range("L12").Value = TextBox32.Value
Sheets("UserIn").Range("L13").Value = TextBox33.Value
Sheets("UserIn").Range("L14").Value = TextBox24.Value

Sheets("UserIn").Range("M5").Value = TextBox35.Value
Sheets("UserIn").Range("M6").Value = TextBox36.Value
Sheets("UserIn").Range("M7").Value = TextBox37.Value
Sheets("UserIn").Range("M8").Value = TextBox38.Value
Sheets("UserIn").Range("M9").Value = TextBox40.Value
Sheets("UserIn").Range("M10").Value = TextBox41.Value
Sheets("UserIn").Range("M11").Value = TextBox39.Value
Sheets("UserIn").Range("M12").Value = TextBox42.Value
Sheets("UserIn").Range("M13").Value = TextBox43.Value
Sheets("UserIn").Range("M14").Value = TextBox34.Value

Load UserForm4
UserForm4.Show
End Sub
```

Figure 13: Code Used to Transfer Resource Inputs from Userform to Worksheet

Clicking on the “next” will then transfer the user to userform 4 that is shown in figure 14. This userform is for the user to enter the productivity conversion factors which represents the ratio between two resources, where one of them is the primary resource and the ratio is the user’s determination of the efficiency of the secondary resource carrying out a task that primarily requires the skills of the primary resource. For example, if resource B were to carry out an activity that has resource A as its primary resource, then the user will enter the efficiency of resource B as a number within the range [0,1] in respect to resource A’s efficiency which would be regarded as 1 as it is the primary resource. If the user were to enter 0.5, then this means that resource B will carry out any task requiring the skill of resource A with a 50% efficient, hence half the productivity of resource A and therefore can conduct the same task independently but in twice the time.

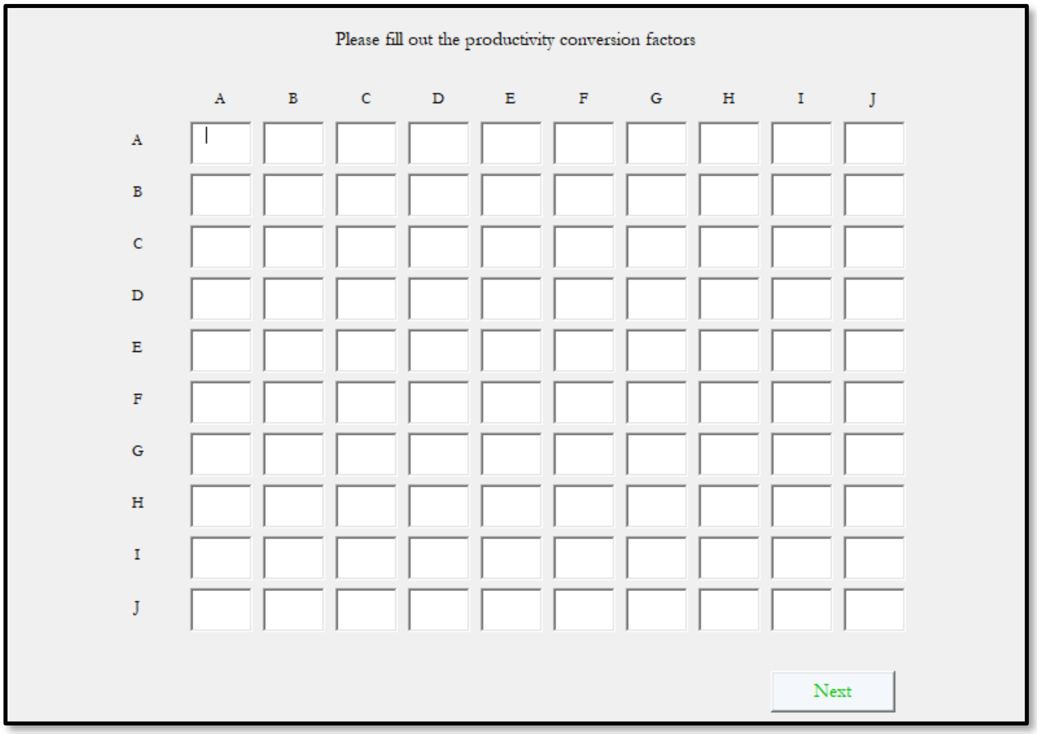


Figure 14: Userform 4 – Resource Productivity Conversion Factors

The code shown in figures 15 and 16 below transfers the information entered by the user from Userform 4 to the User Inputs spreadsheet shown in figure 17.

```
CommandButton4                                  Click

Private Sub CommandButton4_Click()
Unload UserForm4

Sheets("UserIn").Range("AA5").Value = TextBox1.Value
Sheets("UserIn").Range("AA6").Value = TextBox3.Value
Sheets("UserIn").Range("AA7").Value = TextBox6.Value
Sheets("UserIn").Range("AA8").Value = TextBox7.Value
Sheets("UserIn").Range("AA9").Value = TextBox9.Value
Sheets("UserIn").Range("AA10").Value = TextBox10.Value
Sheets("UserIn").Range("AA11").Value = TextBox8.Value
Sheets("UserIn").Range("AA12").Value = TextBox12.Value
Sheets("UserIn").Range("AA13").Value = TextBox13.Value
Sheets("UserIn").Range("AA14").Value = TextBox11.Value

Sheets("UserIn").Range("AB5").Value = TextBox15.Value
Sheets("UserIn").Range("AB6").Value = TextBox16.Value
Sheets("UserIn").Range("AB7").Value = TextBox17.Value
Sheets("UserIn").Range("AB8").Value = TextBox18.Value
Sheets("UserIn").Range("AB9").Value = TextBox20.Value
Sheets("UserIn").Range("AB10").Value = TextBox21.Value
Sheets("UserIn").Range("AB11").Value = TextBox19.Value
Sheets("UserIn").Range("AB12").Value = TextBox22.Value
Sheets("UserIn").Range("AB13").Value = TextBox23.Value
Sheets("UserIn").Range("AB14").Value = TextBox14.Value

Sheets("UserIn").Range("AC5").Value = TextBox25.Value
Sheets("UserIn").Range("AC6").Value = TextBox26.Value
Sheets("UserIn").Range("AC7").Value = TextBox27.Value
Sheets("UserIn").Range("AC8").Value = TextBox28.Value
Sheets("UserIn").Range("AC9").Value = TextBox30.Value
Sheets("UserIn").Range("AC10").Value = TextBox31.Value
Sheets("UserIn").Range("AC11").Value = TextBox29.Value
Sheets("UserIn").Range("AC12").Value = TextBox32.Value
Sheets("UserIn").Range("AC13").Value = TextBox33.Value
Sheets("UserIn").Range("AC14").Value = TextBox24.Value
```

Figure 15: Code to Transfer User Inputs to Worksheet (Part 1)

```
Sheets("UserIn").Range("AD5").Value = TextBox35.Value
Sheets("UserIn").Range("AD6").Value = TextBox36.Value
Sheets("UserIn").Range("AD7").Value = TextBox37.Value
Sheets("UserIn").Range("AD8").Value = TextBox38.Value
Sheets("UserIn").Range("AD9").Value = TextBox40.Value
Sheets("UserIn").Range("AD10").Value = TextBox41.Value
Sheets("UserIn").Range("AD11").Value = TextBox39.Value
Sheets("UserIn").Range("AD12").Value = TextBox42.Value
Sheets("UserIn").Range("AD13").Value = TextBox43.Value
Sheets("UserIn").Range("AD14").Value = TextBox34.Value

Sheets("UserIn").Range("AE5").Value = TextBox45.Value
Sheets("UserIn").Range("AE6").Value = TextBox46.Value
Sheets("UserIn").Range("AE7").Value = TextBox47.Value
Sheets("UserIn").Range("AE8").Value = TextBox48.Value
Sheets("UserIn").Range("AE9").Value = TextBox50.Value
Sheets("UserIn").Range("AE10").Value = TextBox51.Value
Sheets("UserIn").Range("AE11").Value = TextBox49.Value
Sheets("UserIn").Range("AE12").Value = TextBox53.Value
Sheets("UserIn").Range("AE13").Value = TextBox54.Value
Sheets("UserIn").Range("AE14").Value = TextBox52.Value

Sheets("UserIn").Range("AF5").Value = TextBox56.Value
Sheets("UserIn").Range("AF6").Value = TextBox57.Value
Sheets("UserIn").Range("AF7").Value = TextBox58.Value
Sheets("UserIn").Range("AF8").Value = TextBox59.Value
Sheets("UserIn").Range("AF9").Value = TextBox61.Value
Sheets("UserIn").Range("AF10").Value = TextBox62.Value
Sheets("UserIn").Range("AF11").Value = TextBox60.Value
Sheets("UserIn").Range("AF12").Value = TextBox63.Value
Sheets("UserIn").Range("AF13").Value = TextBox64.Value
Sheets("UserIn").Range("AF14").Value = TextBox55.Value

Sheets("UserIn").Range("AG5").Value = TextBox66.Value
Sheets("UserIn").Range("AG6").Value = TextBox67.Value
Sheets("UserIn").Range("AG7").Value = TextBox68.Value
Sheets("UserIn").Range("AG8").Value = TextBox69.Value
Sheets("UserIn").Range("AG9").Value = TextBox71.Value
Sheets("UserIn").Range("AG10").Value = TextBox72.Value
Sheets("UserIn").Range("AG11").Value = TextBox70.Value
Sheets("UserIn").Range("AG12").Value = TextBox73.Value
Sheets("UserIn").Range("AG13").Value = TextBox74.Value
Sheets("UserIn").Range("AG14").Value = TextBox65.Value

Sheets("UserIn").Range("AH5").Value = TextBox75.Value
Sheets("UserIn").Range("AH6").Value = TextBox76.Value
Sheets("UserIn").Range("AH7").Value = TextBox77.Value
Sheets("UserIn").Range("AH8").Value = TextBox78.Value
Sheets("UserIn").Range("AH9").Value = TextBox80.Value
Sheets("UserIn").Range("AH10").Value = TextBox81.Value
Sheets("UserIn").Range("AH11").Value = TextBox79.Value
Sheets("UserIn").Range("AH12").Value = TextBox82.Value
Sheets("UserIn").Range("AH13").Value = TextBox83.Value
Sheets("UserIn").Range("AH14").Value = TextBox44.Value

Sheets("UserIn").Range("AI5").Value = TextBox86.Value
Sheets("UserIn").Range("AI6").Value = TextBox87.Value
Sheets("UserIn").Range("AI7").Value = TextBox88.Value
Sheets("UserIn").Range("AI8").Value = TextBox89.Value
Sheets("UserIn").Range("AI9").Value = TextBox91.Value
Sheets("UserIn").Range("AI10").Value = TextBox92.Value
Sheets("UserIn").Range("AI11").Value = TextBox90.Value
Sheets("UserIn").Range("AI12").Value = TextBox93.Value
Sheets("UserIn").Range("AI13").Value = TextBox94.Value
Sheets("UserIn").Range("AI14").Value = TextBox85.Value

Sheets("UserIn").Range("AJ5").Value = TextBox95.Value
Sheets("UserIn").Range("AJ6").Value = TextBox96.Value
Sheets("UserIn").Range("AJ7").Value = TextBox97.Value
Sheets("UserIn").Range("AJ8").Value = TextBox98.Value
Sheets("UserIn").Range("AJ9").Value = TextBox100.Value
Sheets("UserIn").Range("AJ10").Value = TextBox101.Value
Sheets("UserIn").Range("AJ11").Value = TextBox99.Value
Sheets("UserIn").Range("AJ12").Value = TextBox102.Value
Sheets("UserIn").Range("AJ13").Value = TextBox103.Value
Sheets("UserIn").Range("AJ14").Value = TextBox84.Value

Load UserForm5
UserForm5.Show
End Sub
```

Figure 16: Code to Transfer User Inputs to Worksheet (Part 2)

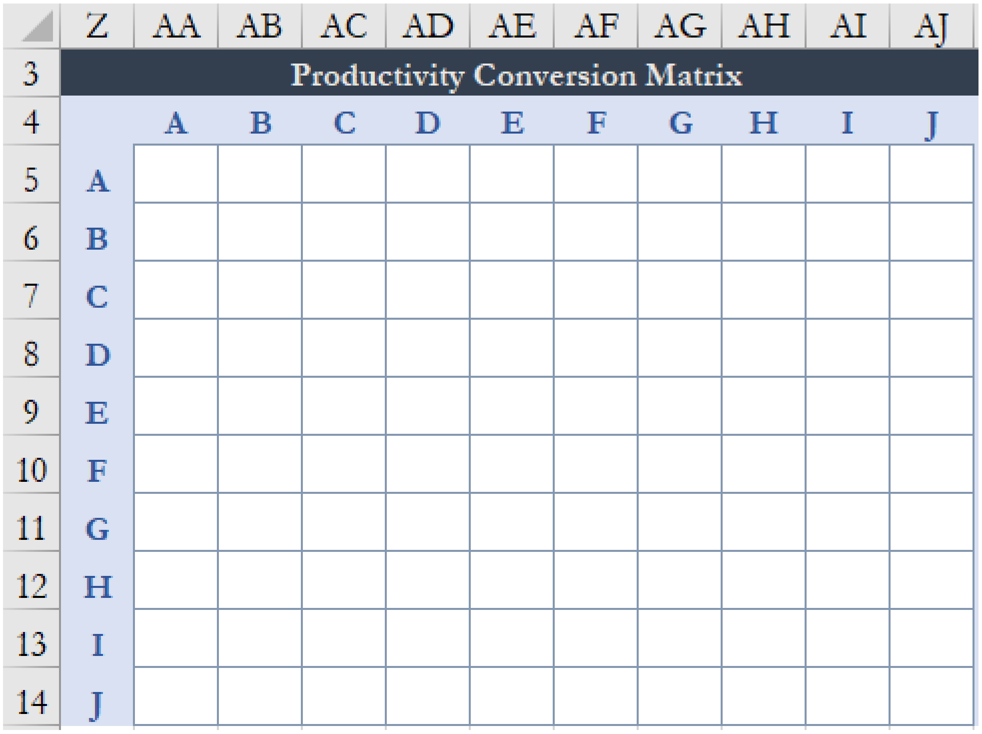


Figure 17: Sample of User Inputs Worksheet Receiving Productivity Factors

The last set of inputs required to be entered by the user are the training costs. The userform shown in figure 18 requests the user to enter the daily wage to be paid to any resource during the multi-skilled training stage.

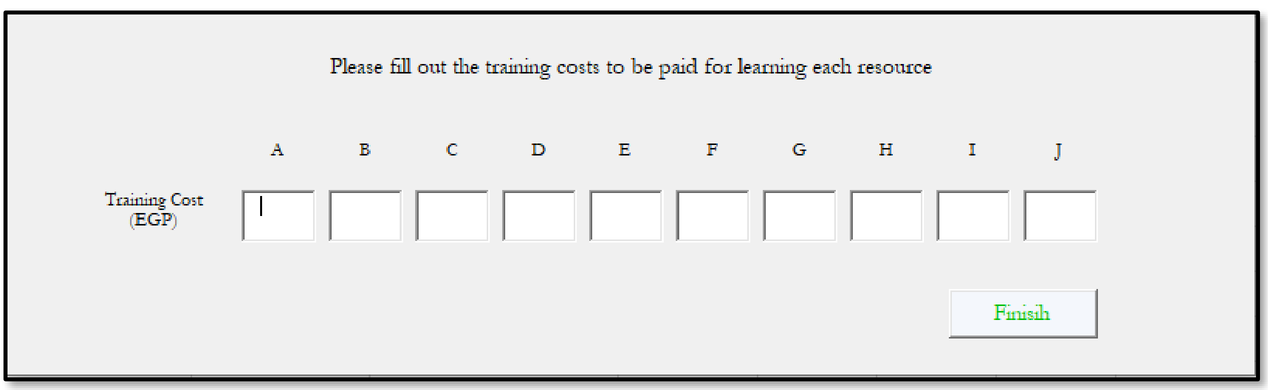


Figure 18: Userform 5 – Training Costs to Learn Each Trade Input

Once all the required information is entered, the model can handover the inputs from the inputs module to the calculations module before optimization can start. Each part of the inputs module is summarized in the next page for a clearer overview of the module.

Activities:

The model allows the input of activities with any number of predecessors, assuming only Finish-to-Start relationships, quantities of each activity, the required primary resource for the completion of each activity. Since the model is generic, it allows the input of activities relevant to any project type.

Resources and Resource Coding:

Upon entering the resources required for each of the activities, the model arranges them in a form, giving each resource a code (A, B, C, etc…) The user also inputs the maximum number of crews available for each of the resources in addition to the daily wage per resource.

Quantities:

Each activity is entered along with its quantity. The quantity is used by the model to determine the change in duration of the activity based on the overall productivity of resources being used to complete the activity.

Resource Productivity and Productivity Matrix:

The user inputs the productivity of each resource. Based on the user's experience or available data, the user specifies the change in productivity for one crew of a primary resource to take on another resource. If resource A were to carry out an activity requiring a primary resource B, and the productivity of A would be 75% of resource B's original productivity, then the input would enter 0.75 in the intersection between A and B. If a resource does not have the toleration or the talent to learn another resource, then the user will input zero in the intersection between the two resources.

3.1.2 Calculations Module

The model has an objective function of minimizing the total project duration. The model would use the input variables, in this case the available crews for each resource. The model constraints comprise of different factors, with the first being limiting the daily resource usage to the maximum available number of crews entered by the user. Another constraint would be to limit the additional number of secondary resources allocated to each activity to avoid crowding on site. Upon running the model, it would be expected that it provides the most optimum option for resource allocation, to provide the user with the lowest project duration, and related costs. The direct labor cost is calculated based on the time each resource is present on site. The time spent on site by each resource is the difference between the last day on site and the first day on site, which is then multiplied by the associated daily wage to give the direct labor cost. The indirect costs directly proportional to the project duration and is calculated by multiplying the project duration by the indirect costs incurred per day. Lastly, the training costs depend on two factors: (1) the number of crews that require training and (2) the trades that they are receiving training for. The summations of those three costs will amount to the total project costs.

3.1.3 Optimization Module

In the optimization module, the model uses Evolver. Evolver is an Excel plug-in that applies the Genetic Algorithms techniques described in *Chapter 2 [Literature Review]* to reach the near-optimum solution. Evolver uses Genetic Algorithms to find the near-optimum solutions for a wide variety of optimization problems, provided that the variables, constraints and objective function are defined. Evolver was first developed and launched in 1989 by Axcelis, Inc. and was the first commercially used GA package that could be used on personal computers. It was then acquired by Palisade Corporations, who updated the program to its current version which is being used for the model's optimization module. The add-in performs several runs by changing the range that is defined as the model "variables". The algorithm keeps running different possibilities in an iterative mode until a near-optimum value of the set objective is reached when the stoppage criteria is met. The stopping criteria could be a user-defined number of run limits or run time.

For this model, the variables were the number of crews, the constraints are the available number of crews, number of crews that could be assigned to each task in addition to the quantity of each activity that needs to be completed. Finally, the objective function is to minimize the project duration. Since the objective function is only to minimize duration, it was integral for the model to also consider the total project cost. The model is meant to minimize duration without risking the increase in project cost, and to do so anther constraint is to be placed into effect. Naturally, indirect costs and training costs both depend on the duration of the project. The longer the project duration, the lower the training costs due to the need of a lower number of crews that will require training, however the higher the indirect cost, in addition to the possibility of imposing liquidated damages if the project were to surpass its given deadline. On the other hand, lower project durations mean that there will be more multiskilling, i.e., more crews that will require training and hence the training costs will increase, however the site indirect costs will decrease. Therefore, for the model to compensate for the unavailability of an objective function to minimize the total project cost, a constraint was added to ensure that the project costs do not increase with reduction in duration. This was done by restricting the training costs to always being less than or equal to the saved indirect costs, and in turn the project's budget will never be exceeded.

3.1.4 Output Module

The output model provides the user with the optimum resource allocation strategy, in addition to specifying and identify to the user which crews will need to learn which skill for the schedule to be carried out as per the optimization model.

3.2 Model Development

3.2.1 Model I – Single Skilled Labor

The single-skilled and multi-skilled labor models are based on the model developed by Abotaleb et al. (2014). This model utilizes the number of crews of the primary resource that is needed for each activity without introducing the usage of multi-skilled labor. The model allows the user to enter the list of activities, the relationship logic between activities, quantities, maximum available crews of each resource and daily wage for each labor resource. The point of this model is to allocate the optimal number of primary resources assigned to each activity. This model is used to generate control results that will be used to compare with the multi-skilled resource model, given that a maximum available budget is assigned by the user. The models will be compared in terms of total project duration and project cost, to determine whether multi-skilled allocation with the introduction of partial allocation is worth being applied or not.

With the use of macro-enabled worksheets, the program allows the user to enter the activities name and ID one by one, with their preceding activities, primary resource and quantities. The next set of Userforms require the user to enter information regarding the resources. For each resource, the user is requested to enter the resource type, resource productivity and maximum number of crews available per day. All these inputs are reflected onto a worksheet named "User Inputs".

The model calculations sheet reads the inputs entered by the user from the "User Inputs" worksheet and reflects them once again into another worksheet, "Model-Single". The resource types are given codes, depending on the trade, to simplify integrating resource types with the model. Table 1 below shows that the first trade specified by the user is given the code "A", then next is given the code "B", etc. Once this table is automatically filed out in the "User Inputs" worksheet, the program converts the resources needed for all activities into code. This is done using a VLOOKUP function as follows:

=VLOOKUP([Cell with Resource entered by User],[The fixed array shown in Table 1 below],2,0)

Table 1: Coding Resources by Trade

Name	Resource Number
Trade 1	A
Trade 2	B
Trade 3	C
Trade 4	D
Trade 5	E

Fixed Array

Once the resources required for each activity are coded, another equation is then used to convert the required primary resource of each activity into matrix form. The need to present the required primary source to complete the activity in matrix form will be further explained later. The equation used for this representation dictates that if for example activity 1 requires resource A, then the number "1" will be placed in the matrix entry intersecting between activity 1 and resource A, otherwise the intersection will be left blank. This is implemented on Microsoft Excel using the IF function below:

=IF('User Inputs'![Required Resource Code] = 'Model- Single'![Fixed Resource Array],1,"")

Table 2: Resource Requirement Matrix

ID	Resource Code	Primary Resource				
		A	B	C	D	E
1	A	1				
2	B		1			
3	B		1			
4	D				1	
5	C			1		
6	B		1			
7	C			1		

Fixed Resource Array

Table 3 below shows the productivity values entered by the user for each of the resources, denoted as P_j. Productivity is a measure of how several units can be complete by one crew of resource per day, which is determined by the user.

Table 3: Productivity Annotation

Name	Resource Number	Productivity
Trade 1	A	P_A
Trade 2	B	P_B
Trade 3	C	P_C
Trade 4	D	P_D
Trade 5	E	P_E

The number of required resources to be assigned for the completion of each activity is given the letter n_i. This represents the model's variables, where iterations are run by the model to determine the best set of values that can provide the near-optimum project duration. The total productivity, P_t, would then be calculated by multiplying the number of crews assigned by the productivity of one crew, as is demonstrated in table 4, and is expressed through the following equation:

$$P_t = n_i \, x \, p_j$$

Where n_i = the number of crews n assigned to activity i

p_j = productivity p of resource j

Table 4: Total Productivity of Resource per Activity

ID	Resource Code	Resource Usage	Total Productivity				
			A	B	C	D	E
1	A	n_1	$n_1.P_A$				
2	B	n_2		$n_2.P_B$			
3	B	n_3		$n_3.P_B$			
4	D	n_4				$n_4.P_D$	
5	C	n_5			$n_5.P_C$		
6	B	n_6		$n_6.P_B$			
7	C	n_7			$n_7.P_C$		

Productivity Array:
Count per activity = 1

The next step comprises of setting a matrix that reflects the maximum available resources that can be used by the activities. This matrix is assigned as one of the constraints to ensure that the resource usage by the activities does not surpass the maximum available number of crews as is shown in Table 5. The matrix was created using the IF function below:

=IF([Primary Resource]=1,[Max Available Value,X_i],0)

Table 5: Maximum Available Resources Matrix

ID	Resource Code	Max. Available	Primary Resource					Max. Availability				
			A	B	C	D	E	A	B	C	D	E
1	A	X_1	1					X_1	0	0	0	0
2	B	X_2		1				0	X_2	0	0	0
3	B	X_3		1				0	X_3	0	0	0
4	D	X_4				1		0	0	0	X_4	0
5	C	X_5			1			0	0	X_5	0	0
6	B	X_6		1				0	X_6	0	0	0
7	C	X_7			1			0	0	X_7	0	0

Another constraint would be to ensure that the count of the resources assigned to each activity is equal to one, since this is the single-skilled resource model, therefore each activity can have only one resource to complete it. This is ensured using the following COUNT function and the constraint is set to “1”:

=COUNTIF([Productivity Array],">0")

Subsequently, knowing the activity quantity, Q_i, which is also specified by the user, the duration of each activity is calculated using this simple equation:

$$d_i = \frac{Q_i}{n_i P_j}$$

The next step would be to determine the direct labor cost required by the activities, depending on the number of days each resource stays on site. This is done by developing a resource bar chart for each of the resources and based on the values of the variable resource numbers, the number of resources required per day can be determined. This is very useful on two accounts, the first being determining the maximum number of days that a resource is available on site (including any idle days in between), and to ensure that the maximum resource limit per day is not exceeded, which is regarded as another constraint.

The total crew days and the number of each resource used per day is determined by adding the number of resource crews required to be used by activities on each day, which was reached using the following SUM function:

$$=SUM([d_n])$$

To find the maximum number of resources, which is constrained to being less than or equal to the maximum available resources per day, the following MAX function is used:

=MAX([Summation of Resources Array])

Where the "Summation of Resources Array" is the total number of allocated resources per day. This is calculated for each individual resource separately.

To determine how many days each resource is present on site, the summation of resources per day array is then summed together. This is the final number that the model uses to determine the total labor cost. This is achieved by multiplying the total number of crew days of each resource by its daily wage, which is specified by the user. The costs of all the resources are added together to determine the total direct labor costs of the project.

The optimization model is run on the objective of minimizing overall project duration.

Objective:

- Minimizing project duration

Variables:

- The number of crews to be used for the completion of each activity

Constraints:

- Maximum resource availability
- Only one primary resource can be assigned per activity

3.2.2 Model II – Multi-skilled Labor

The single-skilled and multi-skilled labor models (discussed in *Section 3.2.1 [Model I – Single-skilled Labor]* and this section) are based on the model developed by Abotaleb et al. (2014). The main difference between the two models is that in this model, any idle labor can be put to work to assist in the completion of another activity, in turn increasing the total productivity of the resources working to complete the activity resulting in a lower duration than if only one resource was assigned to it.

This model uses the exact same inputs that were entered by the user and were used in Model I, in addition to one very integral element, which is the productivity conversion factor matrix. Since the model allows for multi-skilled resource allocation, the drop in productivity of a secondary resource will need to be considered. The user inputs the productivity of each resource. Based on the user's experience or available data, the user specifies the change in productivity, *f*, for one crew of a primary resource to take on another resource. If resource A were to carry out an activity requiring a primary resource B, and the productivity of A would be 75% of resource B's original productivity, then the input would enter 0.75 in the intersection between A and B. If a resource does not have the tolerance or the talent to learn another trade, then the user will input zero in the intersection between the two resources. The user's inputs generate a matrix in the form shown in figure 19 below. This is the conversion matrix which is later used by the calculations module to determine the total productivity of resources per one activity.

Productivity Conversion Factors							
	R_1	R_2	R_3	..	..	..	R_n
R_1	$f_{1,1}$	$f_{1,2}$	$f_{1,3}$	..	..	..	$f_{1,n}$
R_2	$f_{2,1}$	$f_{2,2}$	$f_{2,3}$	..	..	..	$f_{2,n}$
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..
R_n	$f_{n,1}$	$f_{n,2}$	$f_{n,3}$	..	..	..	$f_{n,n}$

Figure 19:Change to matrix with annotations only

The total productivity of resources per activity would be calculated using the following function:

=SUMPRODUCT([f_n], [(Resource Usage Array)$_n$])(Productivity of Primary Resource)*

This can also be expressed mathematically using the equation below:

$$P_t = \sum_{j=1}^{n} n_i \; x \; f_{n,j} \; x \; p_j$$

Where:

$f_{n,j}$ = the productivity conversion factor for secondary resource n carrying out the skill of resource j

A different constraint that would be applied to this model as opposed to model I, would be to ensure that the count of the resources assigned to each activity can be greater than one, since this is the multi-skilled resource model. Therefore, each activity can have more than one resource to complete it. However, a constraint must be added to limit then number of resources that can be assigned to an activity to avoid site crowding. The count of the number of resources per activity is determined using the equation below, and the maximum number of resources constraint is determined by the user, and comes into effect once the iterations are being run by the model:

=COUNTIF([Productivity Array],">0")

Once the total productivity of each activity, P_{nj}, is determined the duration of each activity can easily be determined by dividing its quantity by the total productivity as per the equation below:

$$d_i = \frac{Q_i}{P_{nj}}$$

To determine how several days each resource is present on site, the same concept used in Model I is also used here, where summation of resources per day array is summed together. This is the final number that the model uses to determine the cost per resource. This is done by multiplying the total number of crew days of each resource by its daily wage. The costs of all the resources are added together to determine the total direct cost of the project.

Objectives:

- Minimizing project duration

Variables:

- The number of crews to be used for the completion of each activity

Constraints:

- Maximum resource availability
- A limit of resources per activity as is determined by the user
- The training costs must be less than or equal to the savings in indirect cost.

3.2.3 Model III – Multi-skilled Labor with Partial Allocation

Model III is more complicated than the two previous models, as it introduces the concept of partial allocation, and it is the main contribution of this thesis. As was mentioned in *[Chapter 2 – Literature Review]*, partial allocation was not implemented in any of the research papers that were tackling multi-skilled labor. Partial resource allocation is where a resource is allocated to a part of an activity's duration; whilst in nonpartial resource allocation a resource can only be allocated to the entire activity's duration. Even though this model uses the same user inputs as Models I and II, its calculations module is entirely different.

The first step is creating fields which are later defined as the model variables, which represent the number of crews of each resource assigned per activity fragment. Next, the productivity per part is calculated by multiplying the number of crews per part by the productivity conversion matrix that was introduced in Model II, then multiplying that by the productivity of the primary resource.

The model allows for partial allocation for each day of the duration of any activity and takes into consideration the continuous change in duration per activity with every iteration. Primarily, the model was going to divide each activity into a fixed number of fragments, where each part would simply be the total quantity of the activity divided equally amongst the number of set parts per activity. Each part would then be assigned the optimum number of resources, from which total productivity per part can be achieved. Knowing the productivity per part, duration of each activity would easily be calculated through the following equation:

$$D = \sum_{n}^{i=1} \frac{Q}{TP_i}$$

Where:

Q = Total quantity per activity

T = Number of fragments that activity was divided into

P_i = productivity of resources per part *i*

The problem with this method was that the duration per fragment had to be rounded up to the nearest day, as would be done with any schedule. Traditionally, resources are assigned per day, and it would not be practical to assigned resources per fractions per day. Rounding up of durations to just one day would not significantly affect any schedule, and the final duration would be of a realistic nature. Dividing activities into several parts of the other hand would mean rounding up an n number of days, which could cumulatively increase the project duration to a phantom number that does not in any way represent the activity duration required to complete any set of activities.

Subsequently, an enhanced model was developed to ensure that this same problem would be overcome, and the calculated duration was of higher accuracy. To achieve this, the model was expanded into taking on many more variables, where each activity was not divided into segments that could comprise of several days, but instead each activity was split into divisions of one day. Each activity was allowed a duration of up to 20 days, through which resources can be assigned. The model follows the same concept of model II, however instead of the number of variables equivalent to ten times the number of activities, the number of variables is now twenty-fold, since each activity is not divided into 20 segments.

Table 6 below shows the organization of the model variables. $n_{2,1}R_1$ reflects the number of crews, n, from resource R_1 allocated to activity 2 at day 1 of activity 2. The variables are determined by the model to allocate the optimal number of crews from each resource types during each day of each activity.

Table 6: Partial Allocation Model Variables

ID	R_1					..					R_n				
	1	2	3	..	d_n	1	2	3	..	d_n	1	2	3	..	d_n
1	$n_{1,1}R_1$	$n_{1,1}R_1$	$n_{1,3}R_1$	..	$n_{1,n}R_1$	..	..	..	..	..	$n_{1,1}R_n$	$n_{1,1}R_n$	$n_{1,3}R_n$	..	$n_{1,n}R_n$
2	$n_{2,1}R_1$	$n_{2,2}R_1$	$n_{2,3}R_1$	..	$n_{2,n}R_1$	..	..	..	..	..	$n_{2,1}R_n$	$n_{2,2}R_n$	$n_{2,3}R_n$	..	$n_{2,n}R_n$
3	$n_{3,1}R_1$	$n_{3,2}R_1$	$n_{3,3}R_1$	..	$n_{3,n}R_1$	..	..	..	..	..	$n_{3,1}R_n$	$n_{3,2}R_n$	$n_{3,3}R_n$	..	$n_{3,n}R_n$
4	$n_{4,1}R_1$	$n_{4,2}R_1$	$n_{4,3}R_1$	..	$n_{4,n}R_1$	..	..	..	..	..	$n_{4,1}R_n$	$n_{4,2}R_n$	$n_{4,3}R_n$	..	$n_{4,n}R_n$
5	$n_{5,1}R_1$	$n_{5,2}R_1$	$n_{5,3}R_1$	..	$n_{5,n}R_1$	..	..	..	..	..	$n_{5,1}R_n$	$n_{5,2}R_n$	$n_{5,3}R_n$	..	$n_{5,n}R_n$
6	$n_{6,1}R_1$	$n_{6,2}R_1$	$n_{6,3}R_1$	..	$n_{6,n}R_1$	..	..	..	..	..	$n_{6,1}R_n$	$n_{6,2}R_n$	$n_{6,3}R_n$	..	$n_{6,n}R_n$

The duration of each activity is determined by the assignment of resources. Previously the number of days was calculated by dividing quantity of an activity by the combined productivity of its allocated resources and rounded up to the nearest integer. Alternatively, in this model resources are assigned first, then productivity is calculated based on the number of days that each resource is assigned for. For example, if one resource type will be able to complete part of activity 1 in three days, and a second resource type will be assigned to complete the remaining quantity of the same activity for two days, then the duration of that said activity is the maximum number of days that a resource is assigned, which in this case is three days. The combined productivity of resources per activity is calculated by multiplying two matrices together: (1) a matrix returning 1 and 0 depending on the duration of the activity and (2) the number of deployed crews per activity at each day.

Objectives:

- Minimizing project duration

Variables:

- The number of crews to be used for the completion of each activity

Constraints:

- Maximum resource availability
- A limit of resources per activity as is determined by the user
- Quantity of work that can be completed by the assigned resources must be greater than or equal to the original quantity of each activity
- The training costs must be less than or equal to the savings in indirect cost.

3.2.4 Project Cost

The project costs comprise of the daily wages of the primary and secondary resources, indirect costs, and training costs. The daily wages of the resources are defined by the user and are calculated based on the total man-hours for each resource. The indirect cost per day is also based on the project and is specified by the user. Those costs are regarded as costing measures that are traditionally assessed in any project. The major addition in project costs will be the training cost.

After reaching the near-optimum solution, the model specifies to the user which crews need to learn which skills. This is also used in determining the training costs. For the schedule to be applied as per the reached output, resources will need to learn a specified set of skills. The training costs are determined with the help of the training costs matrix filled out by the user. The matrix identifies the required cost needed to be paid to each resource depending on the skill that will be inducted during the training period. According to the number of crews of each resource that need to attain a new skill, and the training cost defined by the user, the total training cost can be determined and added to the total project cost.

3.3 Model Verification

Verifying the accuracy of any developed model is integral to determine its fitness for the designed purpose. Verification is the act of confirming or refuting the correctness of the proposed algorithm underlying a system in regard to certain measurements using mathematical methods. There are many devised methods that can be used to verify the soundness of developed models, where they aim to answer the question of whether the product conforms to the specifications.

Static verification tests were used to analyze the correctness of the model. This included sampling and correlating measured data and observed test results with calculated expected values to establish conformance with the requirements, and to ensure that model the model produces the expected results with no numerical or behavioral errors. This was done by manually checking each step along the progression of the model development. For example, if the number of resources assigned to a particular activity increases, then the productivity must increase and in turn the duration must decrease, however if the duration increases then the model is not reflecting the correct behavior of its intended use.

This was best represented in the first stages of the development of the multi-skilled with partial allocation model. The model should have produced results where the project duration would be less for the same testing samples that were entered in the multi-skilled labor model that does not allow for partial allocation since assigning more resources to any activity would increase the labor productivity which should in turn reduce duration as a result of the indirectly proportional relationship between duration and productivity given a fixed value for quantity. The model however gave higher durations, as was elaborated on in *section 3.2.3*. Therefore, through verification it was deduced that this method was not generating accurate results, and accordingly another method was devised which then showed correct results.

The verification tests that were implemented were dimensional consistency and extreme conditions. Dimensional consistency checks if all the parameters of an equation are dimensionally consistent. The Extreme Conditions verification test examines the model's response to inputs that are of an extreme nature. If any of the tests failed, then the model would need to be restructured and retested until all the verification tests are satisfied. These tests were carried using hypothetical numbers, and the model was verified by comparing manual calculations with the results that were attained from the models.

Chapter 4 – Case Study and Model Validation

After verifying that the capability of the model to deliver results, its ability to produce accurate results needed to be validated. The validation stage was conducted by testing the model on the same case study that was presented in the research conducted by Abotaleb et.al (2014). Results of this case study were compared with the results attained from the thesis model for validation. The same project data is used for all three models, where the outputs of each model is then compared to show the achieved overall project duration, total cost project cost and cost efficiency. The project parameter and analysis of the model results are herein detailed in this chapter.

4.1 Project Information and Model Inputs

The model was applied to the schedule of a five-story concrete building. The model inputs were entered into the module. As was demonstrated in *Chapter 3 [Model Development]*, multiple Userforms were created to allow the user to enter the inputs listed below:

- Activity name
- Activity ID
- Activity predecessors
- Quantity of activity
- Primary resource
- Productivity of each primary resource
- Productivity conversion factors for secondary resourcing
- List of all resources needed for the project
- Maximum available crews for each resource
- Daily wage for each resource
- Training costs

Upon entering all the activity inputs, the entered data is then transferred to the calculation's module. The activity ID, and predecessors are used to determine the early start and early finish dates of each activity depending on the duration. The case study comprises of 85 activities that represent the following work packages:

- Foundations
- Concrete Works
- Masonry works
- Finishes
- Plumbing
- Electrical works

Accordingly, the required resources were entered and coded as is shown in table 7 below:

Table 7: Resources Needed for the Project

Resource Name	Assigned Code
Carpenter	A
Steel Fixer	B
Insulation Worker	C
Brick Worker	D
Carpenter Plasterer	E
Painter	F
Flooring Worker	G
Plumber	H
Electrician	I

The remaining resource information consists of maximum available resource crews per day, productivity of each resource and the daily wage for each resource. The resource information data of the case study is demonstrated in table 8 below:

Table 8: Resource Information

Resource Code	Productivity	Daily Wage (EGP)	Max No. of Resources per day
A	7.5	100	7
B	0.3	150	6
C	100	150	8
D	25	120	4
E	61	100	2
F	26	120	7
G	15	120	8
H	0.4	100	1
I	8	90	4

A very important input, that is critical for the formulation of multi-skilled labor optimization is the conversion factor matrix. The project manager or user using the program determine the drop in productivity when a skill learns another skill and assists in the completion of a task requiring a secondary skill. The productivity conversion matrix represents the factor that needs to be multiplied by the primary productivity of a primary resource to give the actual productivity of the secondary resource working on that task. Figure 20 shows that if resource G [Flooring Worker] were to participate in completing an activity where resource D [Brick Worker] was its primary resource, then resource G would be working at 50% the capacity of resource D.

Productivity Conversion Matrix									
	A	B	C	D	E	F	G	H	I
A	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0
B	0.6	1	0.5	0.5	0.4	0.6	0	0	0
C	0.4	0.3	1	0	0.3	0.5	0.3	0	0
D	0.6	0.4	0.6	1	0.7	0.7	0.5	0	0.3
E	0.35	0.3	0.5	0.7	1	0.8	0.5	0	0
F	0.4	0	0.5	0.3	0.65	1	0.4	0	0.4
G	0	0	0.8	0.5	0.65	0.6	1	0	0.2
H	0.3	0	0	0.3	0.35	0	0	1	0.7
I	0	0.3	0.3	0	0	0	0.4	0.7	1

Figure 20: Productivity Conversion Rates

The final set of input data was the training costs. Again, depending on experience, the user enters the daily wage of each resource type during their training period. After the model is optimized, another matrix is developed in the calculations module to determine how many crews of each resource need to learn any other skill. The sum product of the matrix in figure 21 below and the developed skills that need to be trained matrix generates the total training costs that will be allocated onto the project's total cost.

Training Cost Matrix									
	A	B	C	D	E	F	G	H	I
A	0	20	20	20	20	20	20	20	20
B	150	0	150	150	150	150	150	150	150
C	50	50	0	50	50	50	50	50	50
D	30	30	30	0	30	30	30	30	30
E	120	120	120	120	0	120	120	120	120
F	70	70	70	70	70	0	70	70	70
G	100	100	100	100	100	100	0	100	100
H	150	150	150	150	150	150	150	0	150
I	200	200	200	200	200	200	200	200	0
J	60	60	60	60	60	60	60	60	60

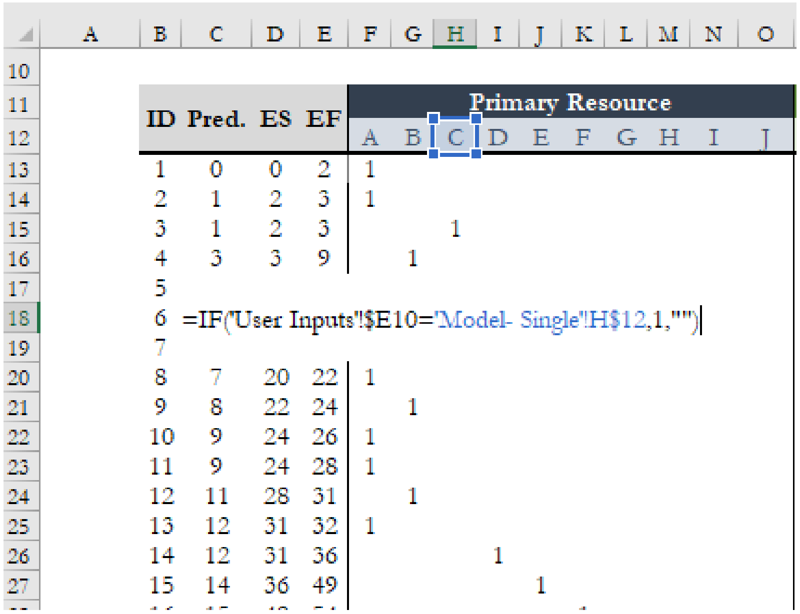
Figure 21: Training Costs Matrix

4.2 Single-skilled Labor Model

The purpose of this model is to use it as a control model to compare the multi-skilled labor allocation models with. The elements that are to be compared are: (1) overall project duration, (2) total project cost, (3) total duration of each resource on site, (4) maximum daily usage reached, (5) hired, working and idle days which will be used to determine (6) expenditure efficiency.

The primary resource inputs are used to create the primary resource matrix, which is used to ensure that each activity uses only one resource type to complete its quantity. The variables must be less than or equal to the maximum available number of crews and must also satisfy the maximum limit of available resources per day which is determined through the resource usage bar charts. When the variable assignments are optimized, the productivity of each resource is calculated by multiplying the number of crews by the productivity of each crew, from which the duration can then be attained by dividing the quantity by total productivity.

Figure 22 shows that the primary resource matrix is based on the resource entry for each activity. The figure demonstrates that if the entered resource for activity 6 is C, then the cell with return a value of “1”, otherwise it will be left blank. The primary resource required for the completion of activity 6 which is foundation insulation works, is an insulation worker, which according to table 9 is coded as resource C (figure 23). Therefore, the selected cell will return a value of “1”.



ID	Pred.	ES	EF	Primary Resource A	B	C	D	E	F	G	H	I	J
1	0	0	2	1									
2	1	2	3	1									
3	1	2	3			1							
4	3	3	9		1								
5													
6	=IF('User Inputs'!$E10='Model- Single'!H$12,1,"")												
7													
8	7	20	22	1									
9	8	22	24		1								
10	9	24	26	1									
11	9	24	28	1									
12	11	28	31		1								
13	12	31	32	1									
14	12	31	36				1						
15	14	36	49					1					

Figure 22: Primary Resource Required by Each Activity

Activity ID	Activity Name	Predeccessors	Required Resources	Resource Code
1	PC footing formwork installation	0	Carpenter	A
2	PC footing formwork removal	1	Carpenter	A
3	RC footing formwork installation	1	Insulation worker	C
4	RC footing steel fixing	3	Steel fixer	B
5	RC footing formwork removal	4	Steel fixer	B
6	Foundation Insulation	5,2	Insulation worker	C
7	Asset el radm	6	Brick worker	D
8	SOG formwork installation	7	Carpenter	A
9	SOG steel fixing	8	Steel fixer	B
10	SOG formwork removal	9	Carpenter	A
11	G floor columns formwork installat	9	Carpenter	A
12	G floor columns steel fixing	11	Steel fixer	B
13	G floor columns formwork remova	12	Carpenter	A

Figure 23: Primary Resource Identification Method

This matrix is particularly important, as it allows for the generation of another matrix that represents the maximum number of resources that can be allocated for each specific activity. Figure 24 for example shows that if the corresponding cell in the first resource matrix is equal to "1", then the selected cell will return the value of maximum available resource, otherwise the cell with be left blank. Figure 25 represents a sample of the user inputs module from which data for calculations is transferred from.

ID	Pred.	ES	EF	Primary Resource										Max. Crews									
				A	B	C	D	E	F	G	H	I	J	A	B	C	D	E	F	G	H	I	J
1	0	0	2	1										7	0	0	0	0	0	0	0	0	0
2	1	2	3	1										7	0	0	0	0	0	0	0	0	0
3	1	2	3			1								0	0	8	0	0	0	0	0	0	0
4	3	3	9		1									0	6	0	0	0	0	0	0	0	0
5	4	9	15		1									0	6	0							
6	5,2	15	16			1								0	0	8	=IF(L18=1,'User Inputs'!$M$11,0)						
7	6	16	20				1							0	0	0							
8	7	20	22	1										7	0	0	0	0	0	0	0	0	0
9	8	22	24		1									0	6	0	0	0	0	0	0	0	0
10	9	24	26	1										7	0	0	0	0	0	0	0	0	0
11	9	24	28	1										7	0	0	0	0	0	0	0	0	0
12	11	28	31		1									0	6	0	0	0	0	0	0	0	0
13	12	31	32	1										7	0	0	0	0	0	0	0	0	0
14	12	31	36				1							0	0	0	4	0	0	0	0	0	0
15	14	36	49					1						0	0	0	0	2	0	0	0	0	0

Figure 24: Identification of Maximum Allowable Number of Crews

	J	K	L	M
3–4	**Resource Number**	**Productivity**	**Daily Wage**	**Max No. of Resources**
5	A	7.5	100	7
6	B	0.3	150	6
7	C	100	150	8
8	D	25	120	4
9	E	61	100	2
10	F	26	120	7
11	G	15	120	8
12	H	0.4	100	1
13	I	8	90	4
14	J		0	0

Figure 25: Maximum Allowable Number of Crews Identification

The maximum number of resources per activity for this model needs to equal one. The column titled "Lim" shown in figure 26 serves the purpose of satisfying the first constraint, by setting the value of all the limits to be equal to one. The limit column is calculated by counting the number of entries greater than zero.

	B	C	D	E	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
10															
11					1	Variables (No. of Crews)									
12	ID	Pred.	ES	EF	Lim	A	B	C	D	E	F	G	H	I	J
13	1	0	0	2	1	7	0	0	0	0	0	0	0	0	0
14	2	1	2	3	1	5	0	0	0	0	0	0	0	0	0
15	3	1	2	3	1	0	0	2	0	0	0	0	0	0	0
16	4	3	3	9	1	0	6	0	0	0	0	0	0	0	0
17	5								0	0	0	0	0	0	0
18	6	=COUNTIF($AA18:$AJ18,">0")							0	0	0	0	0	0	0
19	7								4	0	0	0	0	0	0
20	8	7	20	22	1	7	0	0	0	0	0	0	0	0	0
21	9	8	22	24	1	0	5	0	0	0	0	0	0	0	0
22	10	9	24	26	1	2	0	0	0	0	0	0	0	0	0
23	11	9	24	28	1	5	0	0	0	0	0	0	0	0	0
24	12	11	28	31	1	0	5	0	0	0	0	0	0	0	0
25	13	12	31	32	1	4	0	0	0	0	0	0	0	0	0
26	14	12	31	36	1	0	0	0	3	0	0	0	0	0	0
27	15	14	36	49	1	0	0	0	0	1	0	0	0	0	0

Figure 26: Limitation on Number of Allocated Crews per Resource

The productivity per activity is easily calculated in this model, by simply multiplying the number of crews needed by the productivity of that same resource. An additional matrix was developed, shown in figure 27, to reflect the productivity attributable to each resource per activity. Figure 28 represents a sample of the user inputs module from which data for calculations is transferred from. Since this is a single skilled labor allocation mode, the productivities matrix is of no significance, as the productivity per activity shown in the matrix is equal to the total productivity shown in figure 29 however it was built into the model as it would be a very quick and efficient way to calculate productivities of more than one resource per activity, which will be useful in the calculations of the multi-skilled labor allocation model.

Row	ID	Pred.	ES	EF	Primary Resource A	B	C	D	E	F	G	H	I	J
13	1	0	0	2	1									
14	2	1	2	3	1									
15	3	1	2	3			1							
16	4	3	3	9		1								
17	5	4	9	15		1								
18	6	5,2	15	16			1							
19	7	6	16	20				1						
20	8	7	20	22	1									
21	9	8	22	24		1								
22	10	9	24	26	1									
23	11	9	24	28	1									
24	12	11	28	31		1								
25	13	12	31	32	1									
26	14	12	31	36				1						
27	15	14	36	49					1					

Row	Variables (No. of Crews) A	B	C	D	E	F	G	H	I	J
13	7	0	0	0	0	0	0	0	0	0
14	5	0	0	0	0	0	0	0	0	0
15	0	0	2	0	0	0	0	0	0	0
16	0	6	0	0	0	0	0	0	0	0
17	0	6	0	0	0	0	0	0	0	
18	0	0	2	0	0	0	0	0	0	=IF(H18=1,AC18*'User Inputs'!$K$7,0)
19	0	0	0	4	0	0	0	0	0	
20	7	0	0	0	0	0	0	0	0	0
21	0	5	0	0	0	0	0	0	0	0
22	2	0	0	0	0	0	0	0	0	0
23	5	0	0	0	0	0	0	0	0	0
24	0	5	0	0	0	0	0	0	0	0
25	4	0	0	0	0	0	0	0	0	0
26	0	0	0	3	0	0	0	0	0	0
27	0	0	0	0	1	0	0	0	0	0

Row	New Productivities/unit A	B	C	D	E	F	G	H	I	J
13	52.5	0	0	0	0	0	0	0	0	0
14	37.5	0	0	0	0	0	0	0	0	0
15	0	0	200	0	0	0	0	0	0	0
16	0	1.8	0	0	0	0	0	0	0	0
17								0	0	0
18								0	0	0
19								0	0	0
20	52.5	0	0	0	0	0	0	0	0	0
21	0	1.5	0	0	0	0	0	0	0	0
22	15	0	0	0	0	0	0	0	0	0
23	37.5	0	0	0	0	0	0	0	0	0
24	0	1.5	0	0	0	0	0	0	0	0
25	30	0	0	0	0	0	0	0	0	0
26	0	0	0	75	0	0	0	0	0	0
27	0	0	0	0	61	0	0	0	0	0

Figure 27: Productivity per Resource

Row	Name	Resource Number	Productivity
5	Carpenter	A	7.5
6	Steel fixer	B	0.3
7	Insulation worker	C	100
8	Brick worker	D	25
9	Plasterer	E	61
10	Painter	F	26
11	Flooring worker	G	15
12	Plumber	H	0.4
13	Electrician	I	8
14	-	J	

Figure 28:User Inputs Sheet Where Productivities are Transferred From

	A	B	C	D	E	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV
10																	
11		ID	Pred.	ES	EF	New Productivities/unit										Total Productivity	Duration
12						A	B	C	D	E	F	G	H	I	J		
13		1	0	0	2	52.5	0	0	0	0	0	0	0	0	0	52.5	2
14		2	1	2	3	37.5	0	0	0	0	0	0	0	0	0	37.5	1
15		3	1	2	3	0	0	200	0	0	0	0	0	0	0	200	1
16		4	3	3	9	0	1.8	0	0	0	0	0	0	0	0	1.8	6
17		5	4	9	15	0	1.8	0	0	0	0	0	0	0			5
18		6	5,2	15	16	0	0	200	0	0	0	0	0	0	=SUM(AK18:AT18)		1
19		7	6	16	20	0	0	0	100	0	0	0	0	0			4
20		8	7	20	22	52.5	0	0	0	0	0	0	0	0	0	52.5	2
21		9	8	22	24	0	1.5	0	0	0	0	0	0	0	0	1.5	2
22		10	9	24	26	15	0	0	0	0	0	0	0	0	0	15	2
23		11	9	24	28	37.5	0	0	0	0	0	0	0	0	0	37.5	4
24		12	11	28	31	0	1.5	0	0	0	0	0	0	0	0	1.5	3
25		13	12	31	32	30	0	0	0	0	0	0	0	0	0	30	1
26		14	12	31	36	0	0	0	75	0	0	0	0	0	0	75	5
27		15	14	36	49	0	0	0	0	61	0	0	0	0	0	61	13

Figure 29: Total Productivity

Finally, the duration of each activity is determined by dividing the quantity of each task by the total productivity attained from the deployed resources, as is shown in figure 30. In practice, each activity must be given integer durations, therefore the result was rounded up to the nearest whole number.

	A	B	C	D	E	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW
10																		
11		ID	Pred.	ES	EF	New Productivities/unit										Total Productivity	Duration	
12						A	B	C	D	E	F	G	H	I	J			
13		1	0	0	2	52.5	0	0	0	0	0	0	0	0	0	52.5	2	
14		2	1	2	3	37.5	0	0	0	0	0	0	0	0	0	37.5	1	
15		3	1	2	3	0	0	200	0	0	0	0	0	0	0	200	1	
16		4	3	3	9	0	1.8	0	0	0	0	0	0	0	0	1.8	6	
17		5	4	9	15	0	1.8	0	0	0	0	0	0	0				
18		6	5,2	15	16	0	0	200	0	0	0	0	0	0	=ROUNDUP('User Inputs'!F10/AU18,0)			
19		7	6	16	20	0	0	0	100	0	0	0	0	0				
20		8	7	20	22	52.5	0	0	0	0	0	0	0	0	0	52.5	2	
21		9	8	22	24	0	1.5	0	0	0	0	0	0	0	0	1.5	2	
22		10	9	24	26	15	0	0	0	0	0	0	0	0	0	15	2	
23		11	9	24	28	37.5	0	0	0	0	0	0	0	0	0	37.5	4	
24		12	11	28	31	0	1.5	0	0	0	0	0	0	0	0	1.5	3	
25		13	12	31	32	30	0	0	0	0	0	0	0	0	0	30	1	
26		14	12	31	36	0	0	0	75	0	0	0	0	0	0	75	5	
27		15	14	36	49	0	0	0	0	61	0	0	0	0	0	61	13	

Figure 30: Duration Calculations Sample from Model

Once all the inputs are reflected onto the calculations module of the single-skilled labor model, then constraints, variables and objective function are identified in the model definition window which is accessed through the model. The window is divided into three different partitions. The first is for the identification of the objective function, the second is for selecting the variables and the third is for setting the constraints.

The next and final step would be to calculate the total project cost, which comprises of labor direct costs and indirect costs. The direct labor costs are calculated through the total number of man-days multiplied by the daily wage per each resource (figure 31).

Labor Cost										
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
Total no. of crew days	824	930	182	728	462	612	462	278	306	0
cost/day	100	150	150	120	100	120	120	100	90	0
Total Cost	82400	139500	27300	87360	46200	73440	55440	27800	27540	0
				Total Labor Cost =	**566,980**					

Figure 31: Direct Cost Calculations

Figure 32 shows the model definition window before starting optimization, where the objective function is set to minimize project duration, the variables are the number of crews of each resource for each activity, and the constraints are set to limit one resource to allocated per activity and also to make sure that the number of maximum resource daily usage does not exceed the maximum daily limit set by the user.

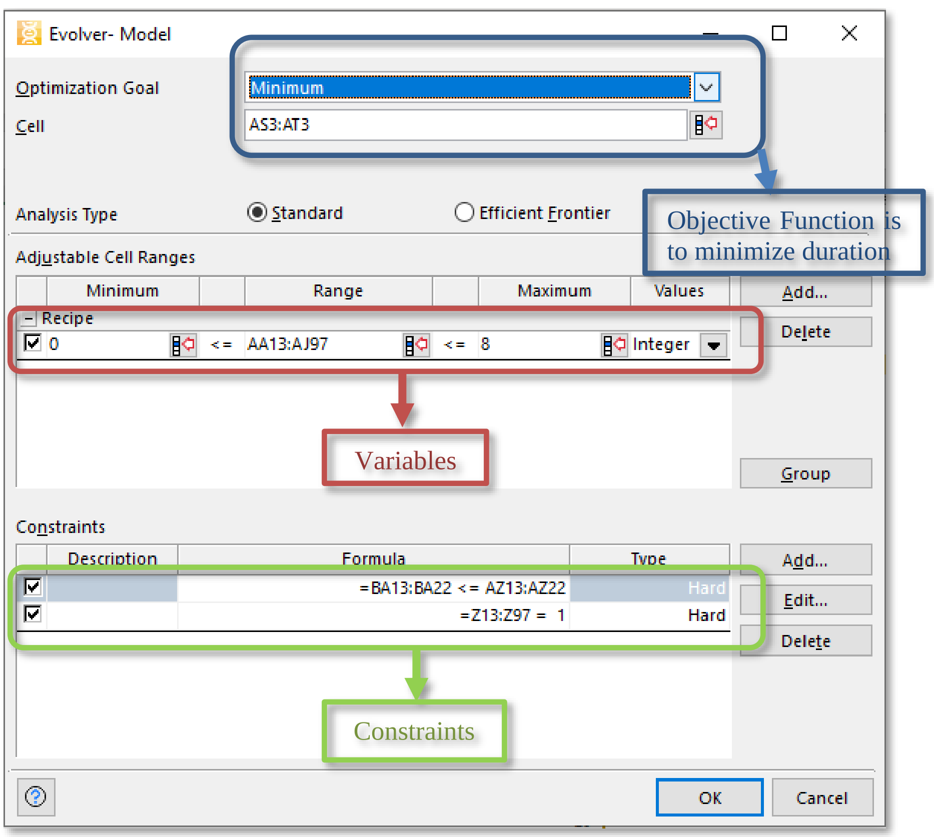


Figure 32: Model Window for Single-Skilled Labor Allocation Model

Figure 33 represents the model outputs (duration and total project cost) of the single-skilled labor allocation model.

Labor Cost

	A	B	C	D	E	F	G	H	I	J
Total no. of crew days	824	930	182	728	462	612	462	278	306	0
cost/day	100	150	150	120	100	120	120	100	90	0
Total Cost	82400	139500	27300	87360	46200	73440	55440	27800	27540	0
					Total Labor Cost =	**566,980**				

Indirect Site Cost

Days	**215**
Cost/day	1000
Total Indirect Cost =	**215,000**
Total Cost	**781,980**

Figure 33:Output of Total Duration and Costs (Single Skilled Labor Model)

The indirect cost was EGP 1000 per day, which meant that the total indirect cost imposed onto the project was EGP 1000 multiplied by the overall project duration. The model was then optimized to determine the optimum assignment of resources to reach the objective function of minimizing both the project duration and total cost. The attained duration was 215 days, with a corresponding indirect cost of EGP 215,000 and direct labor cost of EGP 566,980 making the total project cost equivalent to EGP 781,980.

The model was validated by comparing the duration with the duration that was calculated by Abotaleb et al. (2014) since this model was based on the methodology developed in that paper. The duration attained from the single-skilled labor model that was developed by Abotaleb et al. (2014) was 216 days, which validates the results of the single-skilled labor model developed for this thesis.

Table 9 below gives a summary of the inspected elements in the single-skilled labor allocation model after optimization. The results show that the model is not very cost efficiency having an average expenditure efficiency of 23%, where expenditure efficiency is a measure of how several days the resource is actually doing work relative to the number of days it is hired on site. This means that a large sum is being paid to labors on idle days. Looking at resource C for example, the number of days where the resource was being paid for being on site was 182 days even though the resource was only needed for 19 of those days on site. The multi-skilled labor models aim to increase expenditure efficiency by assigning any resources of their idle days. Expenditure efficiency was calculated using the following equation:

$$Expenditure\ Efficiency\ (\%) = \frac{Working\ Days}{Hired\ Days}\ x\ 100$$

Table 9: Summary of Resource Durations on Site and Expenditure Efficiency

Resource	Hired Days	Working Days	Idle Days	Expenditure Efficiency
A	206	125	81	61%
B	186	42	144	23%
C	182	19	163	10%
D	182	28	154	15%
E	154	30	124	19%
F	153	24	129	16%
G	154	28	126	18%
H	139	34	105	24%
I	153	24	129	16%

ID	Pred.	ES	EF	Primary Resource										Max. Crews										1	Variables (No. of Crews)										New Productivities/unit										Total Productivity	Duration
				A	B	C	D	E	F	G	H	I	J	A	B	C	D	E	F	G	H	I	J	Lim	A	B	C	D	E	F	G	H	I	J	A	B	C	D	E	F	G	H	I	J		
1	0	0	2	1										7	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0	52.5	0	0	0	0	0	0	0	0	0	52.5	2
2	1	2	3	1										7	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0	37.5	0	0	0	0	0	0	0	0	0	37.5	1
3	1	2	3			1								0	0	8	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	200	1
4	3	3	9		1									0	6	0	0	0	0	0	0	0	0	1	0	6	0	0	0	0	0	0	0	0	0	1.8	0	0	0	0	0	0	0	0	1.8	6
5	4	9	15		1									0	6	0	0	0	0	0	0	0	0	1	0	6	0	0	0	0	0	0	0	0	0	1.8	0	0	0	0	0	0	0	0	1.8	6
6	5,2	15	16			1								0	0	8	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	200	1
7	6	16	20				1							0	0	0	4	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	100	4
8	7	20	22	1										7	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0	52.5	0	0	0	0	0	0	0	0	0	52.5	2
9	8	22	24		1									0	6	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0	0	0	0	1.5	0	0	0	0	0	0	0	0	1.5	2
10	9	24	26	1										7	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	15	2
11	9	24	28	1										7	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0	37.5	0	0	0	0	0	0	0	0	0	37.5	4
12	11	28	31		1									0	6	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0	0	0	0	1.5	0	0	0	0	0	0	0	0	1.5	3
13	12	31	32	1										7	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	30	1
14	12	31	36				1							0	0	0	4	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	75	5
15	14	36	49					1						0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	61	0	0	0	0	0	61	13
16	15	49	54						1					0	0	0	0	0	7	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	78	0	0	0	0	78	5
17	10,24	60	62			1								0	0	8	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	100	2
18	15	49	50							1				0	0	0	0	0	0	8	0	0	0	1	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	105	0	0	0	105	1
19	15	49	51									1		0	0	0	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	32	0	32	2
20	15,17	62	72								1			0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.4	0	0	0.4	10
70	67	132	139							1				0	0	0	0	0	0	8	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	30	0	0	0	30	7
71	67	132	139									1		0	0	0	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8	0	8	7
72	67,65	132	142								1			0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.4	0	0	0.4	10
73	65	116	123	1										7	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0	37.5	0	0	0	0	0	0	0	0	0	37.5	7
74	73	123	126		1									0	6	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0	0	0	0	1.5	0	0	0	0	0	0	0	0	1.5	3
75	74	126	132	1										7	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	22.5	0	0	0	0	0	0	0	0	0	22.5	6
76	74	126	131	1										7	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	30	5
77	76	131	134		1									0	6	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0	0	0	0	1.5	0	0	0	0	0	0	0	0	1.5	3
78	77	134	141	1										7	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	7.5	0	0	0	0	0	0	0	0	0	7.5	7
79	77	134	138				1							0	0	0	4	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	100	4
80	79	138	145					1						0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	122	0	0	0	0	0	122	7
81	80	145	148						1					0	0	0	0	0	7	0	0	0	0	1	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	130	0	0	0	0	130	3
82	75	132	133			1								0	0	8	0	0	0	0	0	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	300	1
83	80	145	147							1				0	0	0	0	0	0	8	0	0	0	1	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	105	0	0	0	105	2
84	80	145	152									1		0	0	0	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8	0	8	7
85	80,78	145	155								1			0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.4	0	0	0.4	10

Figure 34: Sample of Single Skilled Labor Calculations Module

4.3 Multi-skilled Labor Model

The multi-skilled labor allocation model follows the exact same techniques and calculations as those demonstrated and explained in Section 4.2 above. The only difference is set in the optimization model definition, where the number of resources per crew limit is altered. Instead of setting the limit to just one resource per activity, the limit is redefined according to the user's preference and ability to their judgement of the optimum number of resources that would avoid overcrowding on site. For this case study the set limit was a minimum of one resource and a maximum of three resources, which means that in addition to the primary resource, a maximum of two other secondary multi-skilled resources can also assist in completing each activity. Just as in the single skilled labor model, the limit column counts the number assigned resources to each of the tasks, which shows that the limit was abided by as is shown in figure 35.

ID	Pred.	ES	EF	Lag	Variables (No. of Crews)										3
					A	B	C	D	E	F	G	H	I	J	1
1	0	0	2	0	7	0	0	0	0	0	0	0	0	0	1
2	1	2	3	0	4	0	0	0	0	0	0	0	0	0	1
3	1	2	3	0	0	0	1	0	0	0	0	0	0	0	1
4	3	3	7	0	0	6	8	0	0	0	0	0	0	0	2
5	4	7	11	0	0	6	8	0	0	0	0	0	0	0	2
6	5,2	11	12	0	0	2	1	0	0	0	0	0	0	0	2
7	6	12	15	0	0	0	3	4	0	0	0	0	0	0	2
8	7	15	17	0	7	0	0	0	0	0	0	0	0	0	1
9	8	17	18	0	0	5	8	0	0	0	0	0	0	0	2
10	9	18	20	0	2	0	0	0	0	0	0	0	0	0	1
11	9	18	20	0	5	4	4	0	0	0	0	0	0	0	3
12	11	20	22	0	4	5	0	0	0	0	0	0	0	0	2
13	12	22	23	0	4	0	0	0	0	0	0	0	0	0	1
14	12	22	27	0	0	0	0	3	0	0	0	0	0	0	1
15	14	27	40	0	0	0	0	0	1	0	0	0	0	0	1
16	15	40	45	0	0	0	0	0	0	3	0	0	0	0	1
17	10,24	48	49	0	0	0	1	0	0	0	1	0	0	0	2

Figure 35: Sample of Resource Allocation after Optimization

Another addition to this model is the conversion productivity factors. The matrix entered using the user's input is integrated into the model as is shown below, where the conversion factors of each resource for each activity are presented. The productivity per resource for each activity is determined by multiplying the conversion factor with the primary resource productivity and multiplied again by the number of crews of the assigned resource, as is shown in figures 36 and 37.

ID	Pred.	ES	EF	Lag	New Productivities/unit											New Productivities/unit									
					A	B	C	D	E	F	G	H	I	J	Prod.	A	B	C	D	E	F	G	H	I	J
1	0	0	2	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	52.5	0	0	0	0	0	0	0	0	0
2	1	2	3	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	30	0	0	0	0	0	0	0	0	0
3	1	2	3	0	0.4	0.3	1	0	0.3	0.5	0.3	0	0	0	100	0	0	100	0	0	0	0	0	0	0
4	3	3	7	0	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0	1.8	1.2	0	0	0	0	0	0	0
5	4	7	11	0	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0	1.8	1.2	0	0	0	0	0	0	0
6	5,2	11	12	0	0.4	0.3	1	0	0.3	0.5	0.3	0	0	0	100	0	60	100	0	0	0	0	0	0	0
7	6	12	15	0	0.6	0.4	0.6	1	0.7	0.7	0.5	0	0.3	0.4	25	0	0	45	100	0	0	0	0	0	0
8	7	15	17	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	52.5	0	0	0	0	0	0	0	0	0
9	8	17	18	0	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0	1.5	1.2	0	0	0	0	0	0	0
10	9	18	20	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	15	0	0	0	0	0	0	0	0	0
11	9	18	20	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	37.5	18	18	0	0	0	0	0	0	0
12	11	20	22	0	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0.72	1.5	0	0	0	0	0	0	0	0
13	12	22	23	0	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	30	0	0	0	0	0	0	0	0	0
14	12	22	27	0	0.6	0.4	0.6	1	0.7	0.7	0.5	0	0.3	0.4	25	0	0	0	75	0	0	0	0	0	0
15	14	27	40	0	0.35	0.3	0.5	0.7	1	0.8	0.5	0	0	0	61	0	0	0	0	61	0	0	0	0	0
16	15	40	45	0	0.4	0	0.5	0.3	0.65	1	0.4	0	0.4	0.3	26	0	0	0	0	0	78	0	0	0	0
17	10,24	48	49	0	0.4	0.3	1	0	0.3	0.5	0.3	0	0	0	100	0	0	100	0	0	0	30	0	0	0

Figure 36: Productivity Conversion Factors

To find the total productivity of resources working on each independent activity, the productivities in figures 37 and 38 above are summed together. The duration is calculated by dividing the quantity of each activity by the total productivity. This is the same technique that was used in model I to calculate overall project duration.

	A	B	C	D	E	F	Q	R	S	T	U	V	W	X	Y	Z	AA
13																	
14		ID	Pred.	ES	EF	Lag	Variables (No. of Crews)										3
15							A	B	C	D	E	F	G	H	I	J	1
16		1	0	0	2	0	7	0	0	0	0	0	0	0	0	0	1
17		2	1	2	3	0	4	0	0	0	0	0	0	0	0	0	1
18		3	1	2	3	0	0	0	1	0	0	0	0	0	0	0	1
19		4	3	3	7	0	0	6	8	0	0	0	0	0	0	0	2
20		5	4	7	11	0	0	6	8	0	0	0	0	0	0	0	2
21		6	5,2	11	12	0	0	2	1	0	0	0	0	0	0	0	2
22		7	6	12	15	0	0	0	3	4	0	0	0	0	0	0	2
23		8	7	15	17	0	7	0	0	0	0	0	0	0	0	0	1
24		9	8	17	18	0	0	5	8	0	0	0	0	0	0	0	2
25		10	9	18	20	0	2	0	0	0	0	0	0	0	0	0	1
26		11	9	18	20	0	5	4	4	0	0	0	0	0	0	0	3
27		12	11	20	22	0	4	5	0	0	0	0	0	0	0	0	2
28		13	12	22	23	0	4	0	0	0	0	0	0	0	0	0	1
29		14	12	22	27	0	0	0	0	3	0	0	0	0	0	0	1
30		15	14	27	40	0	0	0	0	0	1	0	0	0	0	0	1

	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
13																
14	New Productivities/unit											New Productivit				
15	A	B	C	D	E	F	G	H	I	J	Prod.	A	B	C	D	E
16	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	52.5	0	0	0	0
17	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	30	0	0	0	0
18	0.4	0.3	1	0	0.3	0.5	0.3	0	0	0	100	0	0	100	0	0
19	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0	1.8	1.2	0	0
20	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3					
21	0.4	0.3	1	0	0.3	0.5	0.3	0	0	0	100	=(S21*AD21*$AL21)				
22	0.6	0.4	0.6	1	0.7	0.7	0.5	0	0.3	0.4	25					
23	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	52.5	0	0	0	0
24	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0	1.5	1.2	0	0
25	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	15	0	0	0	0
26	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	37.5	18	18	0	0
27	0.6	1	0.5	0.5	0.4	0.6	0	0	0	0	0.3	0.72	1.5	0	0	0
28	1	0.6	0.6	0.5	0.4	0.6	0.6	0	0	0	7.5	30	0	0	0	0
29	0.6	0.4	0.6	1	0.7	0.7	0.5	0	0.3	0.4	25	0	0	0	75	0
30	0.35	0.3	0.5	0.7	1	0.8	0.5	0	0	0	61	0	0	0	0	61

Figure 37: Total Productivity Calculations

ID	Pred.	ES	EF	Lag	New Productivities/unit										Total Productivity	Duration
					A	B	C	D	E	F	G	H	I	J		
1	0	0	2	0	52.5	0	0	0	0	0	0	0	0	0	52.5	2
2	1	2	3	0	30	0	0	0	0	0	0	0	0	0	30	1
3	1	2	3	0	0	0	100	0	0	0	0	0	0	0	100	1
4	3	3	7	0	0	1.8	1.2	0	0	0	0	0	0	0	3	4
5	4	7	11	0	0	1.8	1.2	0	0	0	0	0	0	0	3	4
6	5,2	11	12	0	0	60	100	0	0	0	0	0	0	0	160	1
7	6	12	15	0	0	0	45	100	0	0	0	0	0	0	145	3
8	7	15	17	0	52.5	0	0	0	0	0	0	0	0	0	52.5	2
9	8	17	18	0	0	1.5	1.2	0	0	0	0	0	0	0	2.7	1
10	9	18	20	0	15	0	0	0	0	0	0	0	0	0	15	2
11	9	18	20	0	37.5	18	18	0	0	0	0	0	0	0	73.5	2
12	11	20	22	0	0.72	1.5	0	0	0	0	0	0	0	0	2.22	2
13	12	22	23	0	30	0	0	0	0	0	0	0	0	0	30	1
14	12	22	27	0	0	0	0	75	0	0	0	0	0	0	75	5
15	14	27	40	0	0	0	0	0	61	0	0	0	0	0	61	13
16	15	40	45	0	0	0	0	0	0	78	0	0	0	0	78	5
17	10,24	48	49	0	0	0	100	0	0	0	30	0	0	0	130	1

Figure 38: Sample of Final Durations of Activities

The next and final step would be to calculate the total project cost, which comprises of labor direct costs and indirect costs. The direct labor costs are calculated through the total number of man-days multiplied by the daily wage per each resource (figure 39).

Labor Cost	A	B	C	D	E	F	G	H	I	J
Total no. of crew days	568	660	127	504	444	880	330	384	214	0
cost/day	100	150	150	120	100	120	120	100	90	0
Total Cost	56800	99000	19050	60480	44400	105600	39600	38400	19260	0
Total Primary Labor Cost =						**482,590**				

Figure 39: Direct Cost Calculations

Figure 40 shows model definition before starting optimization, where the objective function is set to minimize project duration, the variables are the number of crews of each resource for each activity. Another set of variables was added to this model, which allowed adding lags before the start of any activities that have float, to postpone their start in order to allow the start of more critical activities by deploying the resources of the delayed activities to be assigned to critical activities. The constraints are set to limit the number of resource to allocated per activity to only three resources, but also to ensure that at least one resource is assigned to that activity. Another constraint was to ensure that the number of maximum resource daily usage does not exceed the maximum daily limit set by the user.

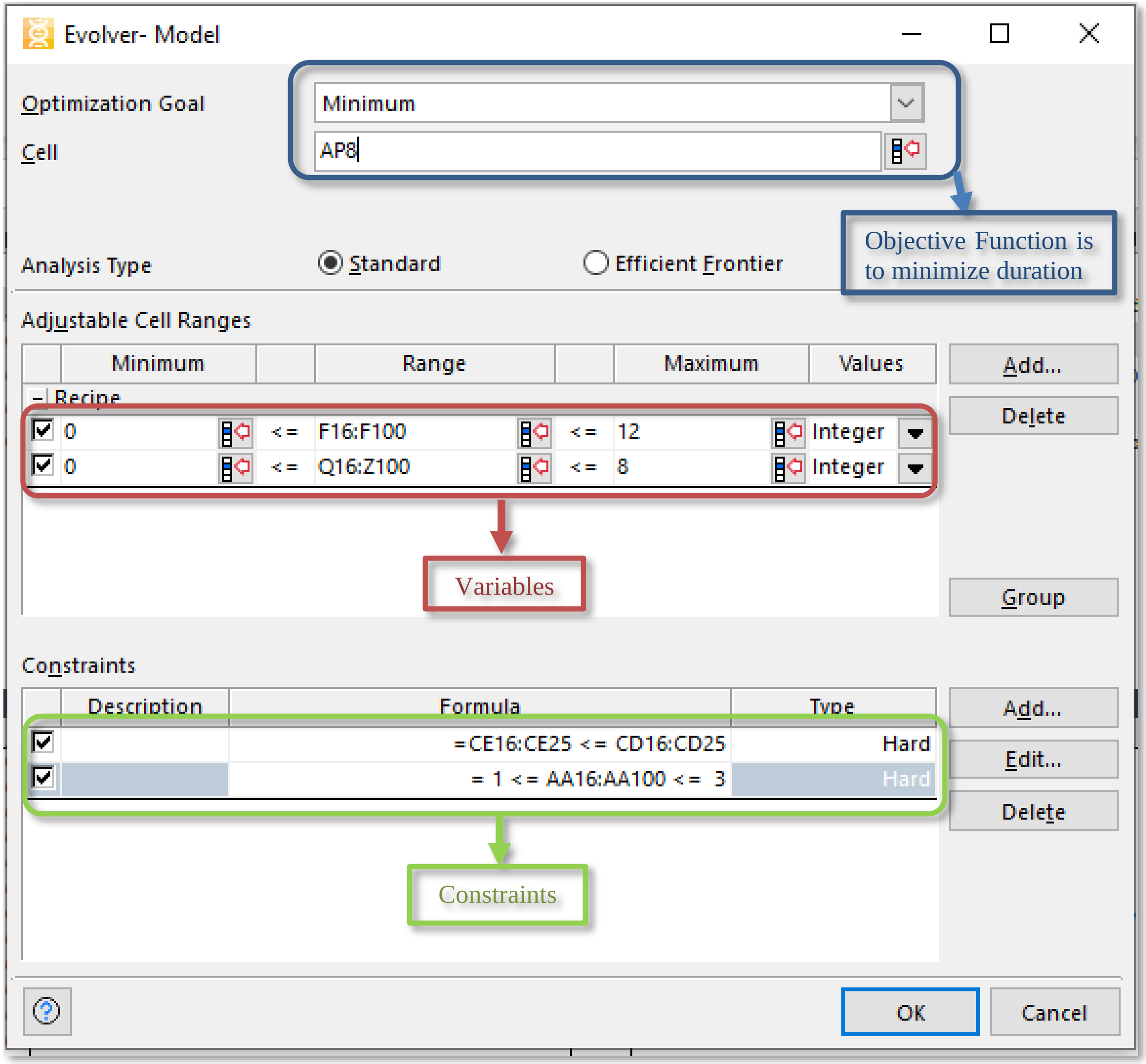


Figure 40: Model Window with the Set Optimization Criteria

Figure 41 represents the model outputs (duration and total project cost) of the multi-skilled labor allocation model.

	Labor Cost										Indirect Site Cost	
	A	B	C	D	E	F	G	H	I	J		
Total no. of crew days	568	660	127	504	444	880	330	384	214	0	Days	148
cost/day	100	150	150	120	100	120	120	100	90	0	Cost/day	1000
Total Cost	56800	99000	19050	60480	44400	105600	39600	38400	19260	0		
Total Primary Labor Cost =	482,590										Total Indirect Cost =	148,000
Training Cost =	16,950										Total Cost	647,540

Figure 41: Output of Total Duration and Costs (Multi-Skilled Labor Model)

The indirect cost was EGP 1000 per day, which meant that the total indirect cost imposed onto the project was EGP 1000 multiplied by the overall project duration. The model was then optimized used Evolver to determine the optimum assignment of resources to reach the objective function of minimizing both the project duration and total cost. The attained duration was 148 days, with a corresponding indirect cost of EGP 148,000 and direct labor cost of EGP 482,590.

The model was validated by comparing the duration with the duration that was calculated by Abotaleb et al. (2014) since this model was based on the methodology developed in that paper. The duration attained from the single-skilled labor model that was developed by Abotaleb et al. (2014) was 148 days, which validates the results of the multi-skilled labor model developed for this thesis.

This model also takes into account the training costs. Training costs are daily wages paid to the workers while in training. The model generated the matrix shown in figure 42 summarizing which resources will help with which skills. This matrix is multiplied by the training costs matrix that is determined by the user, where the product is then multiplied by the total number of induction days. According to literature, training days are usually between 3 to 5 days (Brusco & Johns, 2007) but this can also be determined by the user. The days of induction for this case study were 3 days, giving a training cost of EGP 16,950 making the total project cost equivalent to EGP 647,540.

Training Cost										
	A	B	C	D	E	F	G	H	I	J
A	0	4	4	0	0	0	8	0	0	0
B	4	0	8	0	0	0	0	0	0	0
C	1	2	0	0	1	1	1	0	0	0
D	0	0	3	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0
H	4	0	0	0	0	0	0	0	0	0
I	1	0	1	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

Figure 42: Matrix Used to Calculate Training Costs

Table 10 below gives a summary of the inspected elements in the multi-skilled labor allocation model after optimization. The results show that the model improved the cost efficiency that that of model I, since the expenditure efficiency for the single skilled labor model and the multi-skilled labor model were 23% and 54% respectively, showing an increase of 31% cost efficiency.

Table 10: Summary of Resource Durations on Site and Expenditure Efficiency

Resource	Hired Days	Working Days	Idle Days	Expenditure Efficiency
A	142	97	45	68%
B	132	80	52	61%
C	127	67	60	53%
D	126	83	43	66%
E	111	63	48	57%
F	110	48	62	44%
G	110	29	81	26%
H	96	64	32	67%
I	107	47	60	44%

4.4 Multi-skilled Labor with Partial Allocation Model

The same inputs and methodology as model II is also followed for this model, except for the technique used to determine the duration per activity. To allow partial allocation at any duration of any activity, the activity durations were divided into several parts, which each part representing just one day. The variables in this model are much more, therefore the optimization run time was considerably larger.

Once the resources are assigned, a check is made to make sure that the quantity of each activity has been fulfilled by the assigned resources. The next and final step would be to calculate the total project cost, which comprises of labor direct costs and indirect costs. The direct labor costs are calculated through the total number of man-days multiplied by the daily wage per each resource (figure 43).

Labor Cost

	A	B	C	D	E	F	G	H	I	J
Total no. of crew days	452	588	104	376	156	595	672	74	292	0
cost/day	100	150	150	120	100	120	120	100	90	0
Total Cost	45200	88200	15600	45120	15600	71400	80640	7400	26280	0
				Total Labor Cost =		395,440				

Figure 43: Direct Cost Calculations

Figure 44 shows the model definition before starting optimization, where the objective function is set to minimize project duration, the variables are the number of crews of each resource for each activity. Another set of variables was added to this model, which allowed adding lags before the start of any activities that have float, to postpone their start in order to allow the start of more critical activities by deploying the resources of the delayed activities to be assigned to critical activities. The constraints are set to limit the number of resource to allocated per activity to only three resources, but also to ensure that at least one resource is assigned to that activity. The other two constraints were to ensure that the number of maximum resource daily usage does not exceed the maximum daily limit set by the user, and that the quantity of each activity is covered by the assigned resources. Since this model aims to reduce project duration, it is also important to make sure that reducing project duration does not lead to an increase in project cost. This was done by limiting the training costs to the value of the savings in indirect cost.

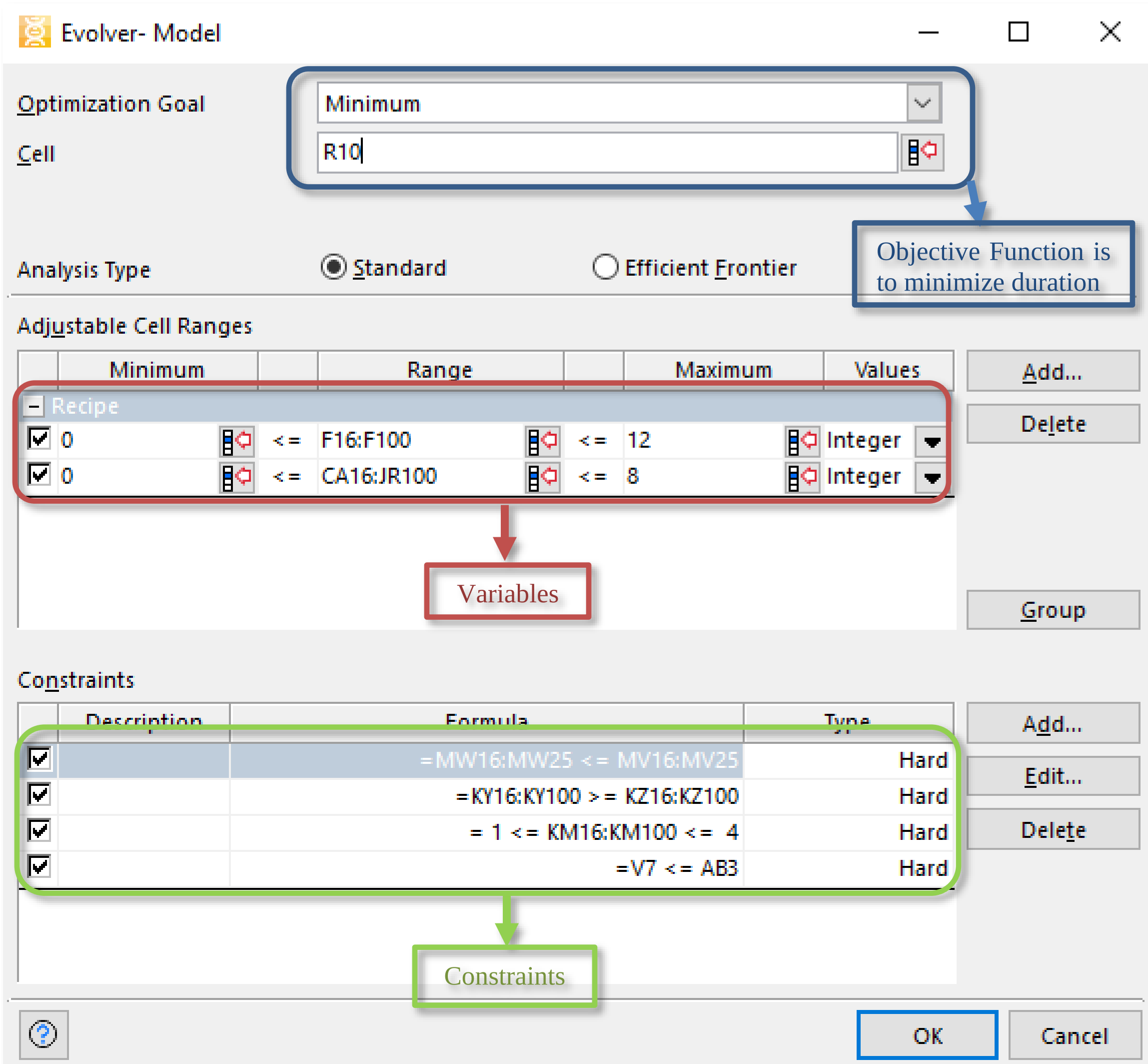


Figure 44: Evolver Window with the Set Optimization Criteria

Figure 45 represents the model outputs (duration and total project cost) of the multi-skilled labor with partial allocation model.

Labor Cost

	A	B	C	D	E	F	G	H	I	J
Total no. of crew days	452	588	104	376	156	595	672	74	292	0
cost/day	100	150	150	120	100	120	120	100	90	0
Total Cost	45200	88200	15600	45120	15600	71400	80640	7400	26280	0
				Total Labor Cost =		395,440				
				Training Cost =		20,400				

Indirect Site Cost

Days	121	Objective: Minimize Duration	Total Indirect Cost	121,000
Cost/day	1000		Total Cost	536,840

Figure 45: Total Duration and Costs (Multi-Skilled Labor with Partial Allocation Model)

The indirect cost was EGP 1000 per day, which meant that the total indirect cost imposed onto the project was EGP 1000 multiplied by the overall project duration. The model was then optimized used Evolver to determine the optimum assignment of resources to reach the objective function of minimizing both the project duration and total cost. The attained duration was 121 days, with a corresponding indirect cost of EGP 121,000 and a direct labor cost of EGP 395,440.

This model also takes into account the training costs. Training costs are daily wages paid to the workers while in training. The model generated the matrix shown in figure 46 summarizing which resources will help with which skills. This matrix is multiplied by the training costs matrix that is determined by the user, where the product is then multiplied by the total number of induction days. The days of induction for this case study were 3 days, giving a training cost of EGP 20,400 making the total project cost equivalent to EGP 536,840.

Training Cost										
	A	B	C	D	E	F	G	H	I	J
A	0	6	8	4	0	7	6	0	0	0
B	7	0	6	0	0	0	0	0	0	0
C	2	1	0	0	0	0	1	0	0	0
D	3	0	4	0	0	0	0	0	1	0
E	0	0	1	0	0	0	1	0	0	0
F	0	0	2	0	0	0	0	0	0	0
G	0	1	2	0	0	1	0	0	0	0
H	0	1	0	0	0	0	2	0	3	0
I	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

Figure 46: Matrix Used to Calculate Training Costs

Table 11 is a generated report that is made available to the user, identifying how several crews of each resource would need to learn which new skill.

Table 11: Identification of No. of Resources Needing to Learn a New Skill

Training Needs								
	6 crews of B need to learn A	8 crews of C need to learn A	4 crews of D need to learn A		7 crews of F need to learn A	6 crews of G need to learn A		
7 crews of A need to learn B		6 crews of C need to learn B						
2 crews of A need to learn C	1 crew of B needs to learn C					1 crew of G needs to learn C		
3 crews of A need to learn D		4 crews of C need to learn D						1 crew of I needs to learn D
		1 crew of C needs to learn E				1 crew of G needs to learn E		
		2 crews of C need to learn F						
	1 crew of B needs to learn G	2 crews of C need to learn G			1 crew of F needs to learn G			
	1 crew of B needs to learn H					2 crews of G need to learn H		3 crews of I need to learn H

Table 12 below gives a summary of the inspected elements in the multi-skilled labor with partial allocation model after optimization. The number of idle days for all resources is less than in models I and II, which in turned led to an increase to the expenditure efficiency of all resources. The average expenditure efficiency of resources was 67%, which is a 44% and 13% increase in expenditure efficiency when compared to the single-skilled and non-partial multi-skilled models, respectively.

Table 12: Summary of Resource Durations on Site and Expenditure Efficiency

Resource	Hired Days	Working Days	Idle Days	Expenditure Efficiency
A	113	82	31	73%
B	98	67	31	68%
C	104	61	43	59%
D	94	73	21	78%
E	78	59	19	76%
F	85	54	31	64%
G	84	27	57	32%
H	74	59	15	80%
I	73	52	21	71%

4.5 Comparison and Analysis of Results

The total number of days on site for each of the nine resources was analyzed and represented in comparison through the following graphs (figures 47 to 55). The graphs represent the average number of deployed resources per week, which also reflects the total number of days were the resource is hired and being paid on site.

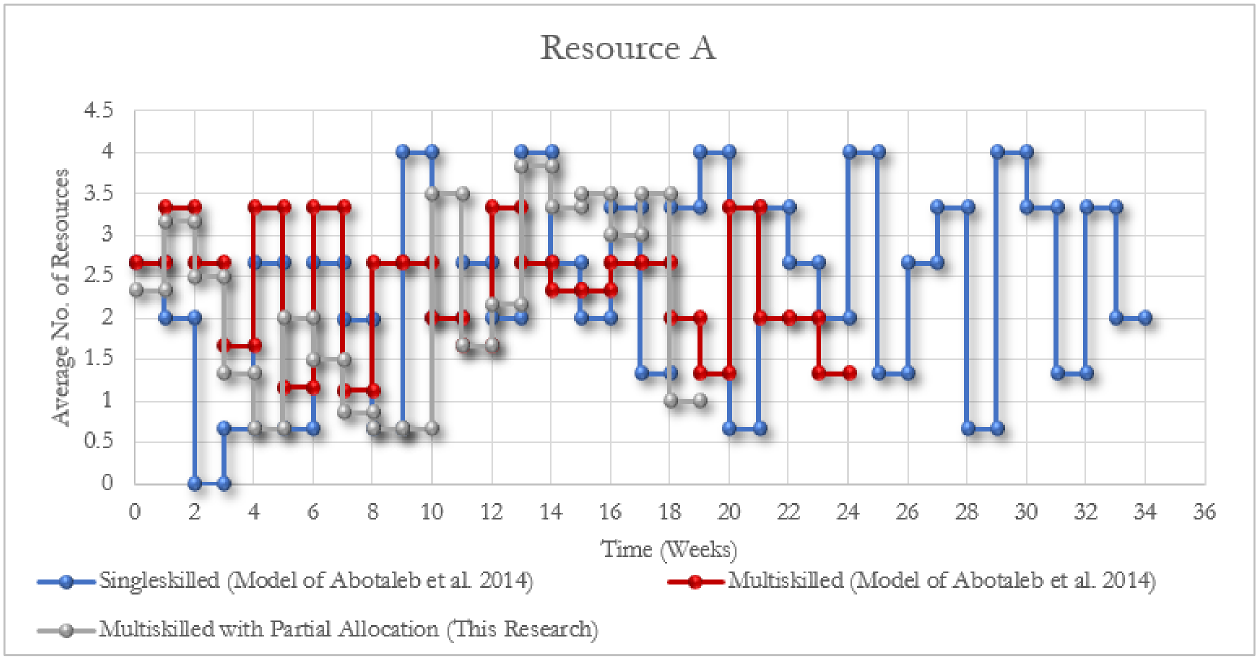


Figure 47: Comparison Between Resource A Usage for All Models

Figure 47 shows that resource A was hired for 34 weeks according to the single-skilled labor allocation model. Since resource A was assigned to activities that start during the first week, the starting date of resource A was always during the first week for all three models. After running model II for multi-skilled allocation, the duration of resource A on site was reduced to 24 weeks (by 29%), which was reduced further to only 19 weeks (by a further 15%) when partial allocation was allowed. The shown trend clearly shows the positive and significant effect of multi-skilled labor optimization, especially when partial allocation is allowed.

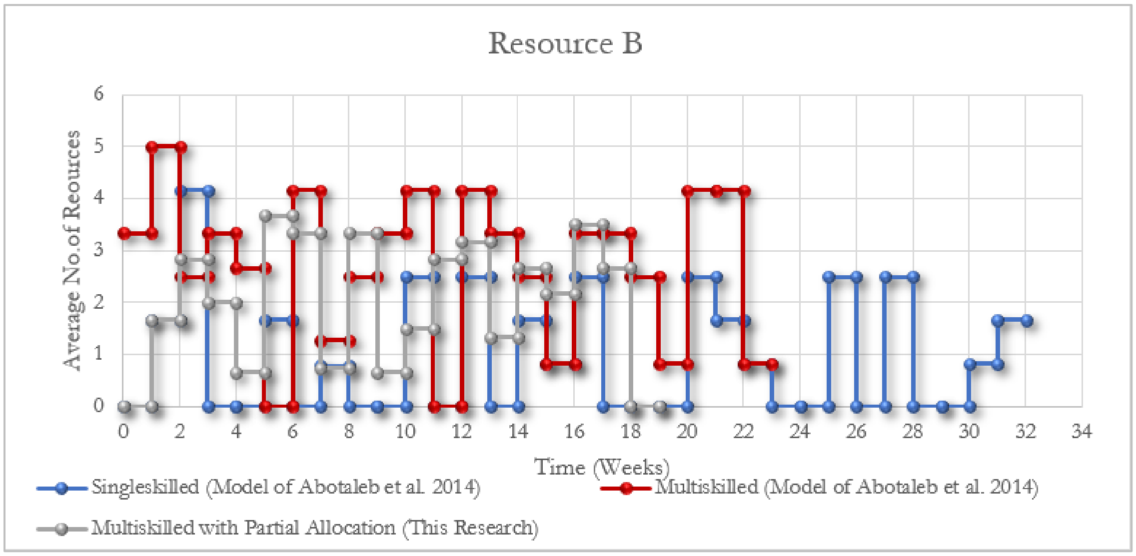


Figure 48: Comparison Between Resource B Usage for All Models

Figure 48 shows that resource B was hired for 32 weeks according to the single-skilled labor allocation model. Since resource B was assigned to activities that also start during the first week, the starting date of resource B was always during the first week for all three models. After running model II for multi-skilled allocation, the duration of resource B on site was reduced to 23 weeks (by 28%), which was reduced further to only 19 weeks (by an additional 12%) when partial allocation was allowed.

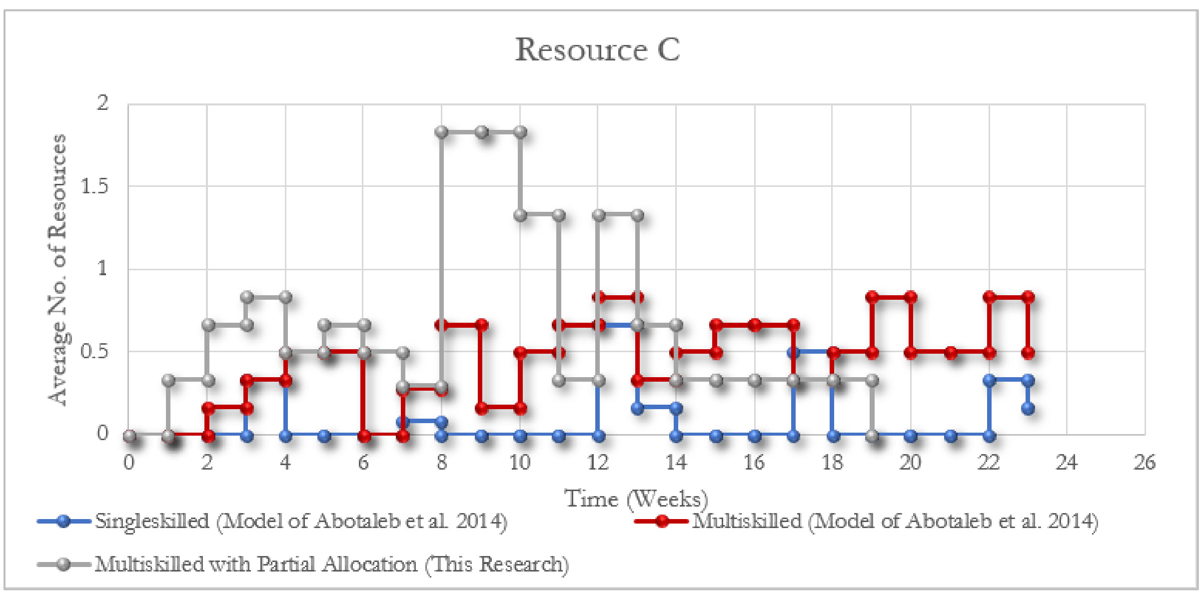


Figure 49: Comparison Between Resource C Usage in All Models

Figure 49 shows that resource C was hired for 23 weeks according to the single-skilled labor allocation model. Since resource C was assigned to activities that also start during the first week, the starting date of resource C was always during the first week for all three models. After running model II for multi-skilled allocation, the duration of resource C on site was still 23 weeks but was reduced to 19 weeks (by 17%) when partial allocation was allowed.

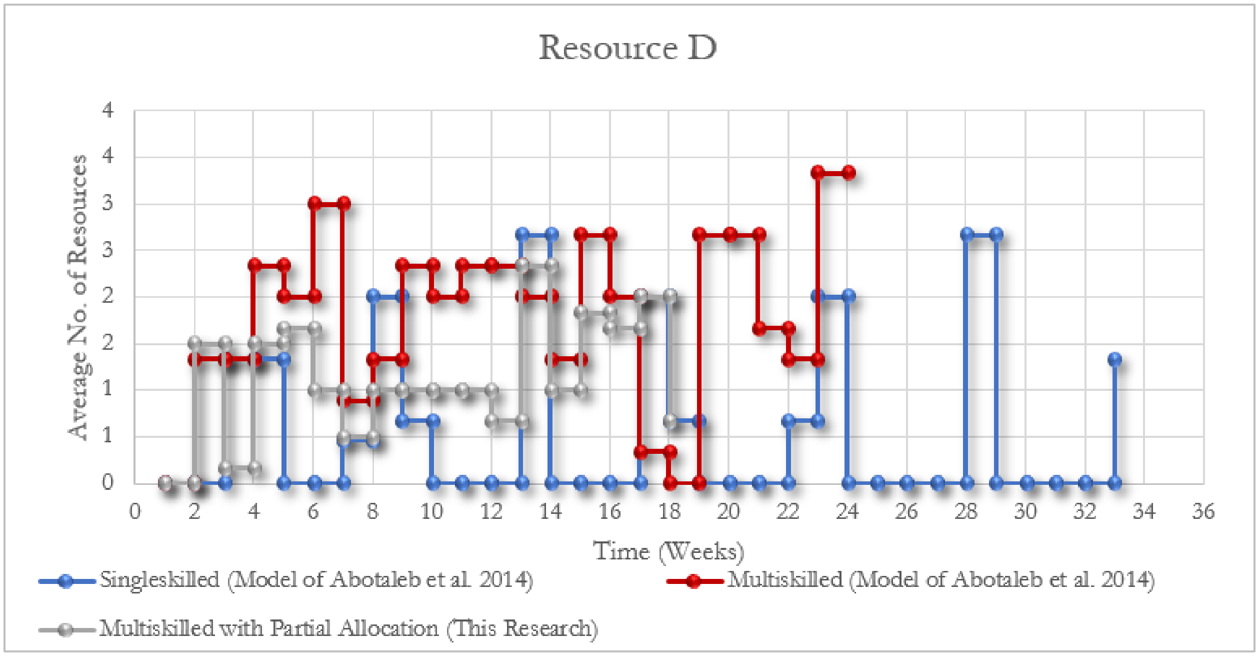


Figure 50: Comparison Between Resource D Usage in All Models

Figure 50 shows that resource D was hired for 34 weeks according to the single-skilled labor allocation model. Resource D started work on week 2 in all the models. This indicates that the preceding tasks of the first activity requiring resource D all ended during or before the start of week 2 of construction. After running model II for multi-skilled allocation, the duration of resource D on site was reduced to 24 weeks (by 29%), which was reduced further to 18 weeks (by an additional 18%) when partial allocation was allowed.

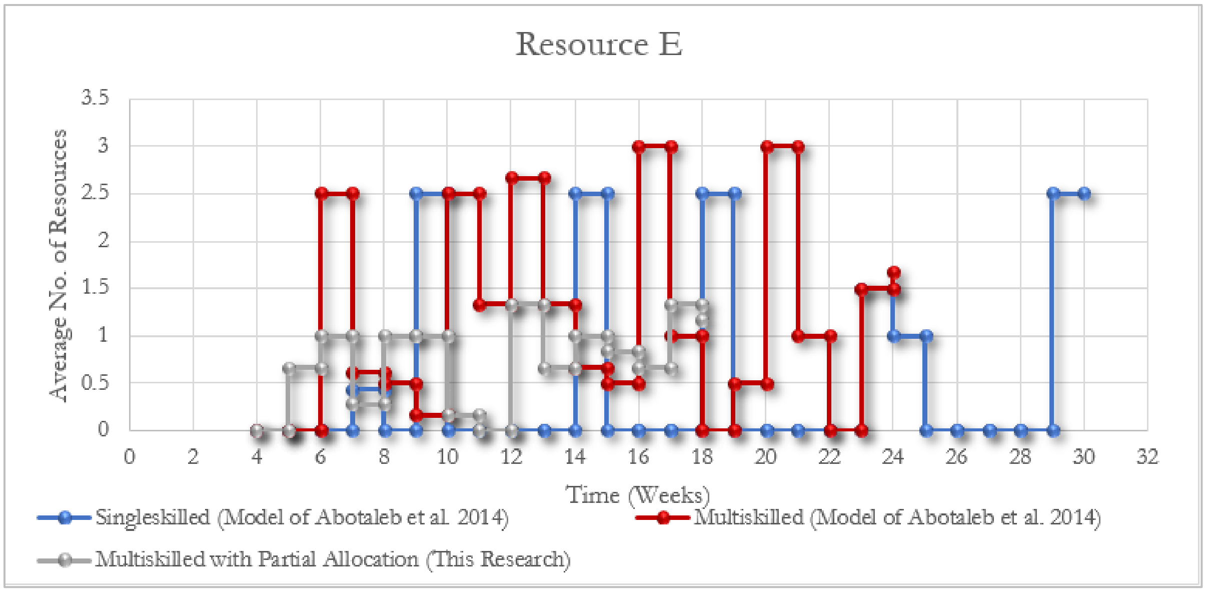


Figure 51: Comparison Between Resource E Usage in All Models

Figure 51 shows that resource E was hired for 30 weeks according to the single-skilled labor allocation model. Resource E started work on week 5 in all the models. This indicates that the preceding tasks of the first activity requiring resource E all ended during or before the start of week 5 of construction. After running model II for multi-skilled allocation, the duration of resource E on site was reduced to 24 weeks (by 20%), which was reduced further to 18 weeks (by an additional 20%) when partial allocation was allowed.

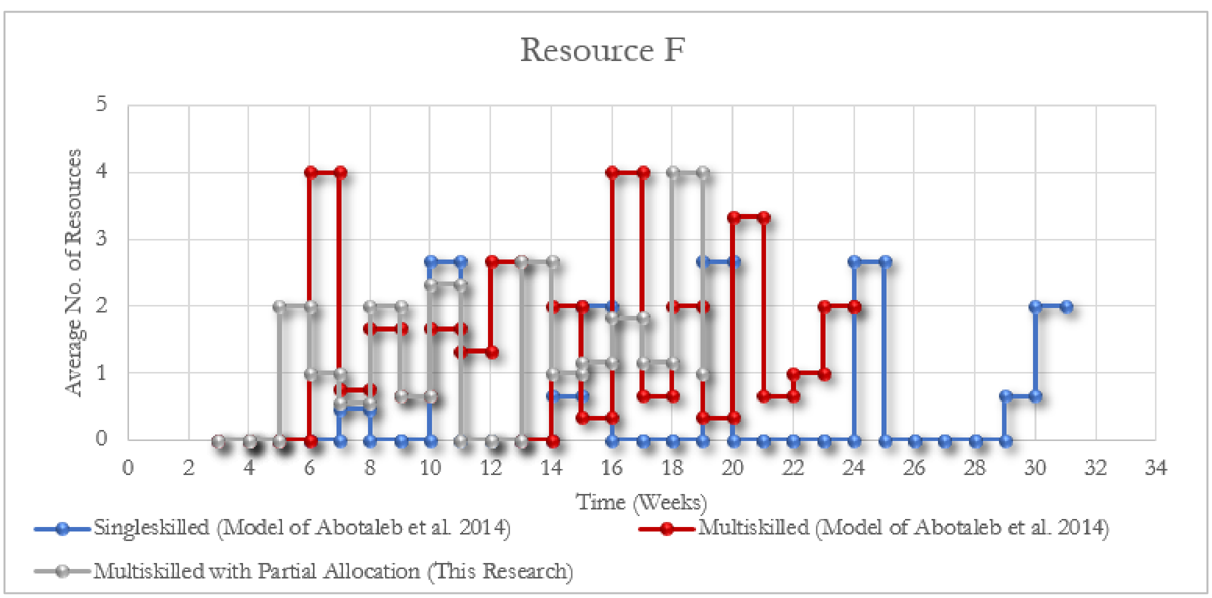


Figure 52: Comparison Between Resource F Usage in All Models

Figure 52 shows that resource F was hired for 31 weeks according to the single-skilled labor allocation model. Resource F started work on week 4 in all the models. This indicates that the preceding tasks of the first activity requiring resource F all ended during or before the start of week 4 of construction. After running model II for multi-skilled allocation, the duration of resource F on site was reduced to 24 weeks (by 22%), which was reduced further to 19 weeks (by an additional 17%) when partial allocation was allowed.

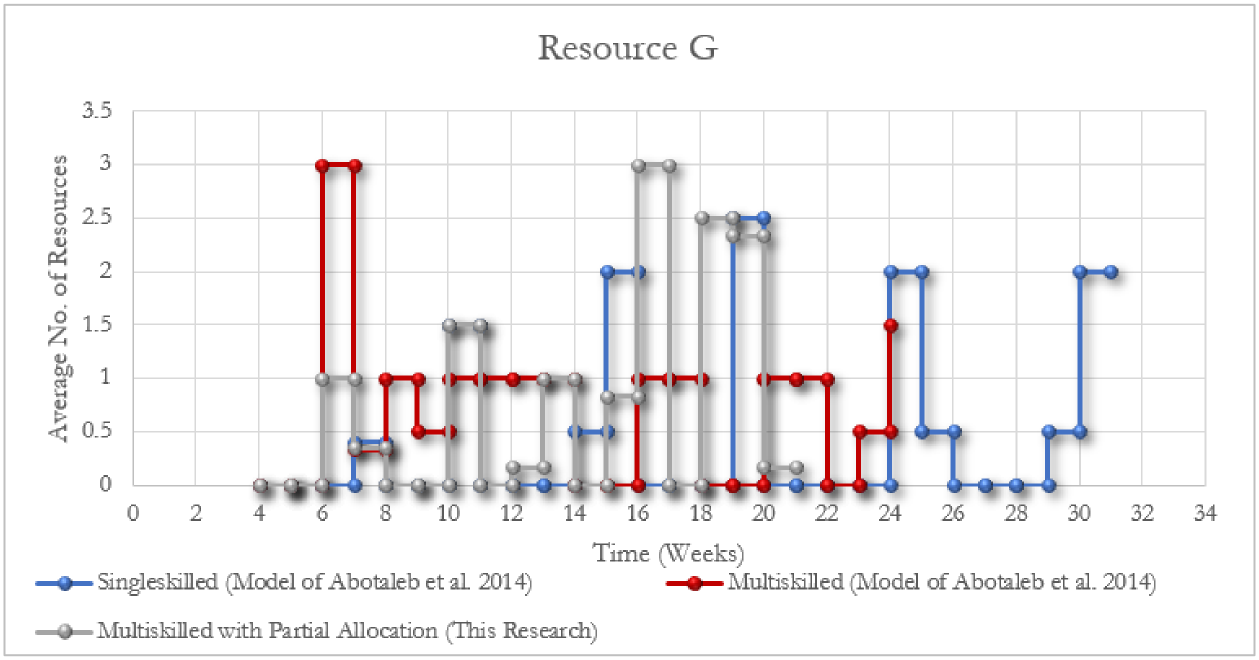


Figure 53: Comparison Between Resource G Usage in All Models

Figure 53 shows that resource G was hired for 31 weeks according to the single-skilled labor allocation model. Resource G started work on week 4 in all the models. This indicates that the preceding tasks of the first activity requiring resource G all ended during or before the start of week 4 of construction. After running model II for multi-skilled allocation, the duration of resource G on site was reduced to 24 weeks (by 22%), which was reduced further to 21 weeks (by an additional 10%) when partial allocation was allowed.

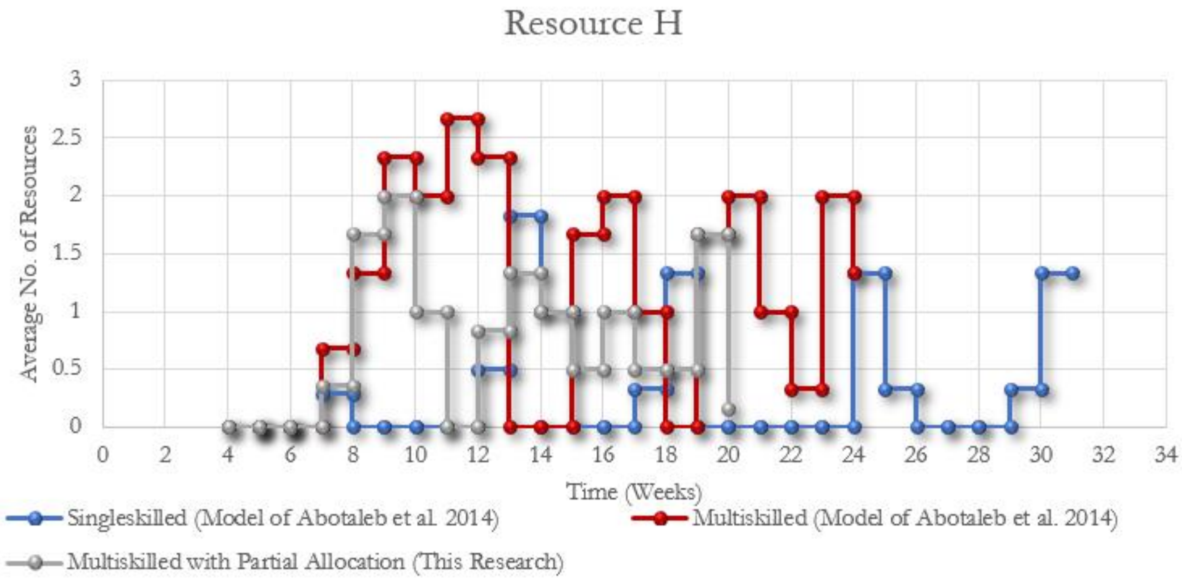


Figure 54: Comparison Between Resource H Usage in All Models

Figure 54 shows that resource H was hired for 31 weeks according to the single-skilled labor allocation model. Resource H started work on week 5 in all the models. This indicates that the preceding tasks of the first activity requiring resource H, all ended during or before the start of week 5 of construction. After running model II for multi-skilled allocation, the duration of resource H on site was reduced to 24 weeks (by 22%), which was reduced further to 20 weeks (by an additional 13%) when partial allocation was allowed.

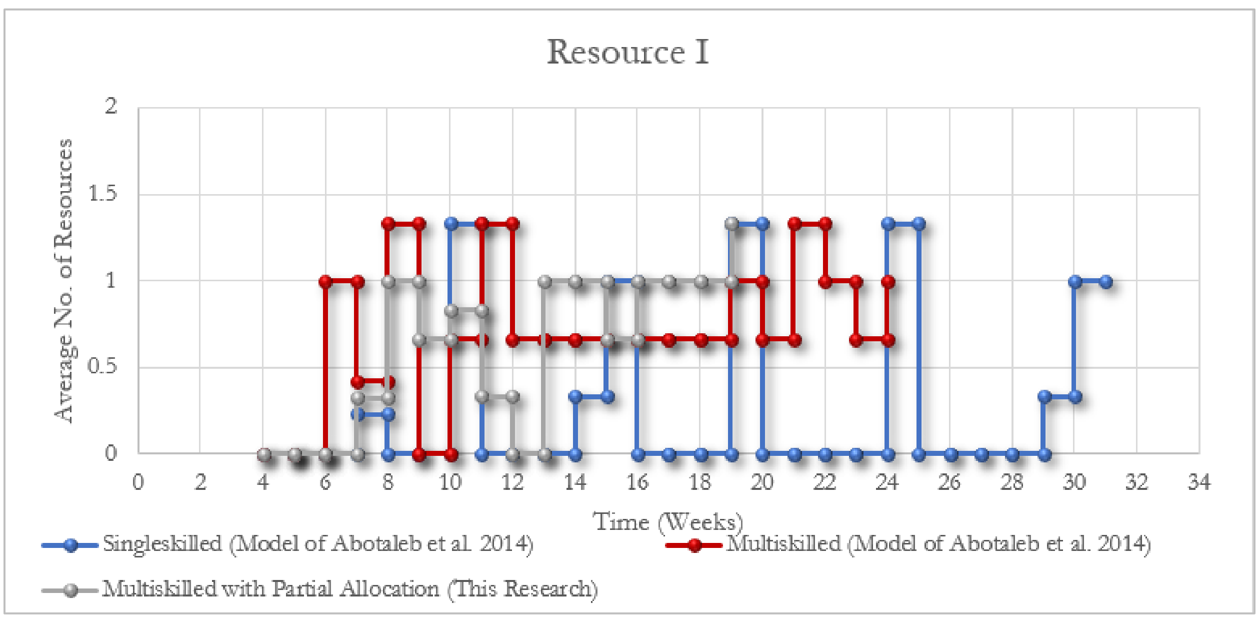


Figure 55: Comparison Between Resource I Usage in All Models

Figure 55 shows that resource I was hired for 31 weeks according to the single-skilled labor allocation model. Resource I started work on week 6 in all the models. This indicates that the preceding tasks of the first activity requiring resource I, all ended during or before the start of week 6 of construction. After running model II for multi-skilled allocation, the duration of resource I on site was reduced to 24 weeks (by 22%), which was reduced further to 19 weeks (by an additional 17%) when partial allocation was allowed.

Table 13 below summarizes the expenditure efficiency results of all nine resources for each of the three models. The results show a clear and significant increase in expenditure efficiency for most of the resources.

Table 13: Comparison of Resource Durations on Site and Expenditure Efficiency

Resource	Hired Days			Working Days			Idle Days			Expenditure Efficiency		
	Single	Multi	Partial	Single	Multi	Partial	Single	Multi	Partial	Single	Multi	Partial
A	206	142	113	125	97	82	81	45	31	61%	68%	73%
B	186	132	98	42	80	67	144	52	31	23%	61%	68%
C	182	127	104	19	67	61	163	60	43	10%	53%	59%
D	182	126	94	28	83	73	154	43	21	15%	66%	78%
E	154	111	78	30	63	59	124	48	19	19%	57%	76%
F	153	110	85	24	48	54	129	62	31	16%	44%	64%
G	154	110	84	28	29	27	126	81	57	18%	26%	32%
H	139	96	74	34	64	59	105	32	15	24%	67%	80%
I	153	107	73	24	47	52	129	60	21	16%	44%	71%

Figure 56 represents the comparison in graph form. Expenditure efficiency measures whether the paid wages are getting any work done in return. It is calculated by finding the percentage of working days per hired days. The higher the expenditure efficiency the better, since that would mean that the resource is put in action for a higher duration of the working days. A low expenditure efficiency means that the number of working days is low, i.e., the number of idle days is high, which is what the model was designed to overcome.

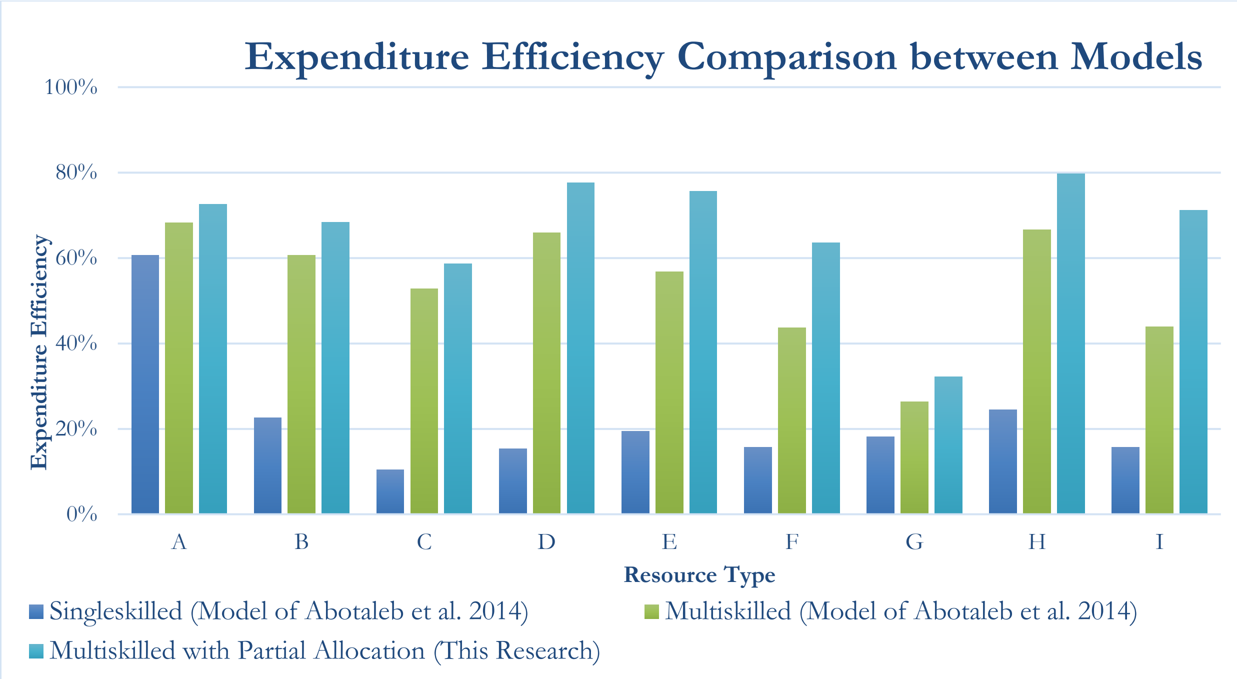


Figure 56: Expenditure Efficiency Comparison for All Models

Table 14 summarizes the duration and cost reduction for all three models. Model III reduced the duration and cost of Model I by 44% and 31% respectively, and this reduced cost also includes the training costs. Modell III also reduced the duration and total project cost attained from Model II but 16% and 18% respectively.

Table 14: Total Duration and Cost for Each Model

Model	Duration (Days)	Labor Cost	Indirect Cost	Training Cost	Total Cost
Single-Skilled[1]	208	566,980	215,000	-	781,980
Multi-Skilled[2]	147	482,590	148,000	16,950	647,540
Multi-Skilled with Partial Allocation[3]	123	395,440	121,000	12,240	528,680
Percentage Reduction[1 to 3]	41%				32%
Percentage Reduction[2 to 3]	16%				18%

1: represents the outputs of the single-skilled labor model (Abotaleb et l. 2014),

2: represents the outputs of the multi-skilled labor model with non-partial allocation of resources (Abotaleb et l. 2014),

3: represents the outputs of the multi-skilled labor model with partial allocation of resources (This research).

Chapter 5 – Conclusion and Recommendation

5.1 Summary and Conclusion

This research presented the methodology of a model that optimizes the allocation of multi-skilled labor that allows for partial allocation. The model was validated by being applied to the activities of a five-storey construction building. The model comprises of three sub-models: (1) a model that allowed only the allocation of one resource type per activity, (2) a model that allowed the allocation of multiple resources to any activity but only for the whole duration of the activity, and (3) a model that assigned multi-skilled labor to any activity at any part of the activity duration. The novelty of this model is that it allowed partial allocation, which was discussed in literature but never applied to any optimization models.

The results showed an increase in expenditure efficiency and noticeable reductions in duration and cost when compared to the single-skilled labor model, which were reduced further when partial allocation was allowed, which proved that the multiskilling with partial allocation concept was a success and would be of great benefit if applied to construction projects scheduling. The model also informs the user of the number of crews of each resource that will need induction to learn a new skill and specifies exactly which skills need to be attained.

5.2 Outcomes and Contributions

The outcomes and contributions that this research has reached in terms of resource allocation can be listed as follows:

- Investigating techniques to cut down on unnecessary project costs
- Introducing a new method of implementing multi-skilled labor that allows for partial allocation
- Contributing to the increase in productivity of site tasks by putting idle labor into use to reduce project duration which in turn reduces the associated indirect site costs
- Identifying the number of resources that would need to learn another certain skill.

When applied a case study was applied to the model, it generated positive results, where the reduction in duration between the single skilled allocation and multi-skilled mode with labor allocation was 44%. The project overall cost was also reduced by 31%.

5.3 Limitations and Recommendations for Future Research

The main limitation that was faced was the running time of the multi-skilled labor with partial allocation model as it had to run thousands of iterations before it could reach the optimum solution. The number of variables would increase more with a higher number of activities, and therefore longer processing time would be required. Therefore, it is recommended that a different heuristic method be used for the investigation of the effects of multi-skilled labor on construction projects. Examples of such methods would be ant colony or shuffled-frog leaping.

Another recommendation would be regarding the crews that need to receive training. The model was able to identify the number of resource crews that need to receive induction and for which skill. It would be better if each of the resource crews be identified separately and integrated with the model to specify to the user exactly which crews need to receive training for which skill. This would also help determine exactly when each crew leaves and enters the site, which would represent the actual situation on site, to determine the exact labor costs needed depending on the number of days each crew is on site.

REFERENCES

Abotaleb, I., Moussa, M. & Hussain, S., 2014. *Optimization of Allocating Multi-Skilled Labor Resources Using Genetic Algorithms.* Halifax, NS, CSCE 2014 General Conference.

Ahmadian Fard Fini, A., Akbarnezhad, A., Rashida, T. & Waller, T., 2017. Job Assignment Based on Brain Demands and Human Resource Strategies. *Journal of Construction Engineering and Management*, 143(5).

Ahmadian Fard Fini, A., Rashidi, T., Akbarnezhad, A. & Travis-Waller, S., 2016. Incorporating Multiskilling and Learning in the Optimization of Crew Composition. *Journal of Construction Engineering and Management,* 142(5), pp. 401-409.

Al-Bazi, A. & Dawood, N., 2010. Developing Crew Allocation System for the Precast Industry Using Genetic Algorithms. *Computer Aided Civil and Infrastructure Engineering,* 25(8), pp. 581-595.

Al-Bazi, A. & Dawood, N., 2017. Simulation-based Optimization Using Simulated Annealing for Crew Allocation in the Precase Industry. *Architectural Engineering and Design Management,* 10(1), pp. 1-18.

Alster, N., 1989. What Flexible Workers Can Do. *Fortune,* 199(4), pp. 62-66.

Arashpour, M. et al., 2017. Optimal Process Integration Architectures in Off-site Construction: Theorizing the Use of Multi-skilled Resources. *Architectural Engineering and Design Management,* 14(1-2), pp. 46-59.

Arashpour, M. et al., 2016. Off-site Construction Optimization: Sequencing Multiple Job Classes with Time Constraints. *Automation in Construction,* 71(2), pp. 262-270.

Arashpour, M., Wakefield, R., Blismas, N. & Minas, J., 2015. Optimization of Process Intregration and Multi-skilled Resource Utilization in Off-site Construction. *Automation in Construction,* Volume 50, pp. 72-80.

Arditi, D., Tokdemir, O. B. & Suh, K., 2001. Effect of Learning on Line of Balance Scheduling. *International Journal of Project Management,* 19(5), pp. 265-277.

BCA, B. a. C. A., 2016. *Multi-skilling Scheme for Construction Industry.* s.l.:Building and Construction Authority.

Bhoyar, S. & Parbat, D., 2014. Optimal Scheduling for Repetitive Construction Projects with Mutliple Resource Crews. *International Journal of Emerging Technology and Advanced Engineering,* 4(6), pp. 302-207.

Bittel, R. & Trepo, G., 1979. The Starting-up of a New Plant Organized in Multi-skilled Production Groups. *Working on the Quality of Working Life.*

Brusco, M. & Johns, T., 2007. Staffing a Multiskilled Workforce with Varying Levels of Productivity: An Analysis of Cross-training Policies. *Decisions Sciences.*

Burleson, R. C., Haas, C. T., Tucker, R. I. & Stanley, A., 1989. Multiskilled Labor Utillization Strategies in Construction. *Journal of Construction Engineering and Management,* 124(6), pp. 480-489.

Cai, X. & Li, K. N., 2000. A Genetic Algorithm for Scheduling Staff of Mixed Skills Under a Multi-Criteria. *European Journal of Operational Research,* 125(2), pp. 359-369.

Carley, L. A., Haas, C., Borcherding, J. & Goodrum, P., 2003. Experiences with Multiskilling Among Non-union Craft Workers in US Industrial Construction Projects. *Engineering, Construction and Architectural Management,* 10(6), pp. 374-381.

Carmicheal, D. G., 2005. *Project Management Framework.* s.l.:CRC Press.

Cross, M., 1986. Multiskilling Brings Cost and Productivity Benefits. *Process Engineering,* 72(11), pp. 51-53.

El-Gafy, M. A., 2006. *Resource Allocation in Repititive Construction Schedules using Ant Colony Optimization.* s.l., Joint International Conference.

Ettlie, J. & Rezo, E. M., 1992. Organization, Integration and Process Innovation. *Journal of Academic Management,* Volume 35, pp. 795-827.

Florez, L., 2017. Crew Allocation System for the Masonry Industry. *Computer-Aided Civil and Infrastructure Engineering,* 32(10), pp. 874-889.

Francis Siu, M. F., Lu, M. & Abourizk, S., 2015. *Methodology for Crew-job Allocation Optimization in Project and Workface Scheduling.* s.l., s.n., pp. 652-659.

Goldberg, D. E., 1989. *Genetic Algorithm in Search, Optimization and Machine Learning.* Boston, MA: Addison-Wesley Longman Publishing Co., Inc..

Gomar, J., 1999. *Assignment and Allocation of a Partially Multiskilled Workforce,* Austin, Texas: s.n.

Gomar, J., Haas, C. & Morton, D., 2002. Assignment and Allocation Optimization of Partially Multi-skilled Workforce. *Journal of Construction Engineering and Management,* 128(2), pp. 103-109.

Gouda, A., Hosny, O. & Nassar, K., 2017. Optimal Crew Routing for Linear Repetitive Projects using Graph Theory. *Automation in Construction,* Volume 81, pp. 411-421.

Haas, C., Rodriquez, A. M., Glover, R. & Goodrum, P., 2001. Implementing a Multiskilled Workforce. *Construction Management and Economics,* 19(6), pp. 633-641.

Hegazy, T., Shabeeb, A. K., Elbeltagi, E. & Cheema, T., 2000. Algorithm for Scheduling with Multi-skilled Constrained Resources. *Journal of Construction Engineering and Management,* 126(6), pp. 414-421.

Holland, J. H., 1975. *Adaptation in Natural and Artifical Systems: An Introductory Analysis with Aplications to Biology, Control and Artificial Intelligence.* Ann Arbor: University of Michigan Press.

Ho, P., 2016. Labour and Skill Shortages in Hong Kong's Construction Industry. *Engineering, Construction and Architectural Management,* 23(4), pp. 533-550.

Hottenstein, M. & Bowman, S., 1998. Cross-training and Worker Flexibility: A Review of DRC System Research. *The Journal of High Technology Management Research,* 9(2), pp. 157-174.

Hyari, K., El-Mashaleh, M. & Kandil, A., 2010. Optimal Assignment of Multiskilled Labor in Building Construction Projects. *International Journal of Construction Education and Research,* Volume 6, pp. 70-80.

Lee, D.-E., Lim, T.-K. & Arditi, D., 2011. An Expert System for Auditing Quality Management Systems in Construction. *Computer-Aided Civil and Infrastructure Engineering,* 26(8), pp. 612-631.

Lill, I., 2009. Multiskilling in Construction a Strategy for Stable Employment. *Technological and Economic Development of Economy,* 15(4), pp. 540-560.

Liu, C. & Yang, S., 2011. A Hybrid Genetic Algorithm for Integrated Project Task and Multi-skilled Workforce Scheduling. *Journal of Computational Information Systems*, 7(6), pp. 2187-2194.

Liu, M. & Ortsman, O., 1979. Developing New Forms of Work Organization in a New Chemical Plant in France. *Working on the Quality of Working Life.*

Liu, S.-S. & Wang, C.-J., 2011. *An Optimization Model for Linear Project Scheduling Considering Multi-skilled Crews.* Seoul, ISARC.

Liu, S.-S. & Wang, C.-J., 2012. Optimizing Linear Project Scheduling with Multi-skilled Crews. *Automation in Construction,* Volume 24, pp. 16-23.

Liu, S.-S. & Wang, C. J., 2007. Optimization Model for Resource Assignment Problems of Linear Construction Projects. *Automation in Construction,* 16(4), pp. 460-473.

Long, L. D. & Ohsato, A., 2009. A Genetic Algorithm-based Method for Scheduling Repititve Construction Projects. *Automation in Construction,* Volume 18, pp. 499-511.

MOM, M. o. M., 2016. *Recognising Workplace Safety and Health (WSH) Competencies.* s.l.:Ministry of Manpower.

Moselhi, O. & Alshibani, A., 2007. Crew Optimization in Planning and Control of Earthmoving Operations using Spacial Technologies. *ITcon,* 12(2), pp. 121-137.

Nallikari, M., 1995. Multi-skilled Work Teams. *J. Ship Production,* 11(4), pp. 239-244.

Nasirian, A., Arashpour, M. & Abbasi, B., 2019. Critical Literature Review of Labor Multiskilling in Construction. *Journal of Construction Engineering and Management,* 145(1), pp. 1-13.

Nassar, K., 2005. Evolutionary Optimizatin of Resource Allocation in Repititive Construction Schedules. *ITcon,* 10(1), pp. 265-273.

Nonaka, I., 1990. Redundant, Overlaping, Organizations, A Japanese Approach to Managing the Innovation Process. *California Management Review,* Volume 605, pp. S5-S8.

Ofori, G., 2002. Singapore's Construction: Moving Toward a Knowledge-based Industry. *Building Research & Information,* 30(6), pp. 401-412.

Pagano, S. & Heathcote, K., 2003. *Education of Self Taught Building Contractors.* Reston, VA, ASCE, pp. 1-7.

Pandey, P. & Maheswari, J. U., 2015. *Improvised Scheduling Framework Integrating WS, MS, & DS for Repititive Construction Projects.* Oulu, Finland, International Association for Automation and Robotics in Construction.

Pike, R., 1986. *Optimization for Engineering Systems.* New York: Van Nostrand Reinhold.

Pollitt, D., 2010. Construction Companis Build Skills for the Future. *Training and Management Development Methods,* 24(1), pp. 515-518.

Project Management Institute , 2017. *A Guide to the Project Management Body of Knowledge.* 6 ed. Netwon Square(PA): Project Management Instiute.

Qin, R., Memghard, D. A. & Barnes, W., 2015. Workforce Flexibility in Operations Management. *Surveys in Operations Research and Management Science,* 20(1), pp. 19-33.

Shahhosseini, V. & Sebt, M. H., 2011. Competency-based Selection and Assignment of Human Resources to Construction Projects. *Scientia Iranica,* 18(2), pp. 163-180.

Srour, I., Haas, C. & Morton, D., 2006. Linear Programming Approach to Optimize Investment in the Construction Workforce. *Journal of Construction Engineering and Management,* 132(11), pp. 1158-1166.

Tam, C., Tong, T. & Cheung, S. O., 2001. Genetic Algorithm Model in Optimizing the Use of Labour. *Construction Management and Economics,* 19(2), pp. 207-215.

Tatum, D. J., 1989. Organizing to Increase Innovation in Construction Firms. *Journal of Construction Engineering and Management ASCE*, 115(4), pp. 602-17.

Treleven, M., 1989. A Review of the Dual Resource Constrained System Research. *IIE Transactions,* 21(3), pp. 279-287.

Wang, Y., Goodrum, P., Haas, C. & Glover, R., 2009. Analysis of Observed Skill Affinity Patterns and Motivation for Multiskilling among Craft Workers in the U.S. Industrial Construction Sector. *Journal of Construction Engineering and Management*, 135(10), pp. 999-1008.

Wongwai, N. & Malaikrisanachalee, S., 2010. Augmented Heuristic Algorithm for Multi-skilled Resource Scheduling. *Elsevier,* 20(4), pp. 429-445.