DRAFT DOCUMENT

MASTER

QUANTITATIVE METHODS FOR DECISION-MAKING IN ECONOMICS AND BUSINESS

MASTER´S FINAL WORK

PROJECT

SCHEDULING MULTI-SKILLED TASKS AND ENGINEERS

TELECOMMUNICATION COMPANY'S PROBLEM

RAIMUNDO TRAÇA MARQUES SOARES RAMOS

SUPERVISION:
PROF. DR. MARIA CÂNDIDA VERGUEIRO MONTEIRO CIDADE MOURÃO
LUÍS LEITÃO, SCOTLAND CONTRACTS MANAGER

OCTOBER – 2021

GLOSSARY

AP – Assignment Problem

BLP – Binary Linear Programming

FTTH – Fibre to the Home

MFW – Master's Final Work

MILP – Mixed Integer Linear Programming

MSPSP – Multi-Skill Project Scheduling Problem

SP – Schedule Problem

TTSP – Technicians and Tasks Schedule Problem

VBA – Visual Basic for Applications

Resumo

O corrente projeto contribui para o agendamento de tarefas e de engenheiros com múltiplas qualificações. Este foi desenvolvido no âmbito de uma empresa de telecomunicações escocesa – Locktel Ltd – e surge da necessidade sentida na empresa de, automaticamente e com precisão, agendar tarefas e alocar engenheiros.

Para resolver o problema foram desenvolvidos dois modelos de programação linear (um misto e um binário) interligados. O primeiro agenda as tarefas e atribui-lhes um engenheiro – líder. O segundo identifica as tarefas, entre as já agendadas, que necessitam de mais engenheiros – parceiros – e atribui-os. O objetivo é maximizar as qualificações, requeridas pelas tarefas, dos engenheiros afetos a tarefas. A decisão de desenvolver dois modelos ajuda a abordar o problema de maximização identificado em situações variadas. Além disso, os parceiros só podem ser atribuídos quando os líderes já são conhecidos, sendo que cada engenheiro designado no primeiro modelo se torna um líder, no segundo modelo, e os restantes potenciais parceiros.

Ambos os modelos foram implementados num ficheiro Excel, através de linguagem VBA, e resolvidos, com métodos exatos, usando o *OpenSolver*. Tanto as leituras de *inputs* como a folha de *output* do programa Excel foram construídas de modo a corresponderem na sua plenitude ao *layout* do programa já utilizado pela Locktel, o BigChange.

Por fim, após a análise dos resultados obtidos, foi possível averiguar uma potencial melhoria das qualificações dos engenheiros escolhidos com o uso do programa desenvolvido. Adicionalmente, foi interessante observar a relação inversa entre o número de tarefas a realizar num só dia e a respetiva média das qualificações dos engenheiros escolhidos, realçando o crescente número de compromissos com o aumento no número de tarefas por agendar num só dia.

PALAVRAS-CHAVE: *Technicians and Tasks Scheduling Problem;* Programação Linear; *Visual Basic for Applications*; *OpenSolver*

CÓDIGOS JEL: C; C44

ABSTRACT

The current project contributes to the schedule of multi-skilled tasks and engineers. This project was developed in a Scottish telecommunications company framework – Locktel Ltd – and emerges from the necessity to automatically and accurately schedule tasks and assign engineers to those tasks.

In order to address the problem, two linked linear programming models (one mixed and one binary) were developed. The first one schedules all tasks and assigns one engineer per task – leader. The second one identifies which tasks, within the already scheduled ones, need more engineers – mates – and assigns them. The goal is to maximise the tasks' required qualifications of the allocated engineers. The decision to develop two models helps to address the identified maximisation problem in various situations. Moreover, mates can only be found once the leaders are already known, as every engineer assigned in the first model becomes a leader, in the second model, and the remain engineers are potential mates.

The models were written in an Excel file, through VBA language, and solved by exact methods using the OpenSolver. Both inputs and the output sheets were built in order to thoroughly match the layout already used by Locktel's program, the BigChange.

Lastly, after the obtained results' analysis, it was possible to observe a potential improvement in the chosen engineers' qualifications with the help of the developed program. Additionally, it was interesting to see the inverse relation between the number of tasks to schedule in one day and the respective average of the chosen engineers' qualifications, highlighting the increase of compromises that need to be done with the rise of unscheduled tasks in one day.

KEYWORDS: Technicians and Tasks Schedule Problem; Linear Programming; Visual Basic for Applications; OpenSolver

JEL CODES: C; C44

INDEX

List of Figures

List Of Tables

ACKNOWLEDGMENTS

There are many people whom I would like to thank for their contribution. This project represents not only the end of my masters, but also the end of an important phase of my life. Its development would not be possible without the support of family, friends, colleagues, and professors. Therefore, I would like to thank all people that played a relevant role throughout this journey.

First and foremost, I wish to thank Professor Maria Cândida Mourão for her encouragement, creativity and tireless guidance and support.

I am also thankful to my family for their support and patience. To my mother, Carla, for providing me with the opportunity to pursue with my studies and unconditional love. To my sisters, Leonor and Helena, for being there to help me and always believe in me, even when I doubted.

I am also grateful for my colleagues and friends, especially Ana Isabel Ferreira, Margarida Sousa, and Raquel Nunes for numerous discussions.

Finally, I would like to thank Locktel and its representatives. Thank you Luís Leitão, Pedro Santos, and Nuno Vieira for helping me understand the world of telecom with your knowledge. Thank you, Jozef Farkas and Peter Vaskanin, for the confidence vote and for receiving me so well in Locktel's headquarters.

1. INTRODUCTION

This project arose in the context of a Master's in Quantitative Methods for Decision-making in Economics and Business final work, at Lisbon School of Economics and Management (ISEG), Universidade de Lisboa (UL). The project consists in the development and implementation of a Mixed Integer Linear Programming (MILP) and a Binary Linear Programming (BLP), modelled to allocate multi-skilled engineers to multi-skill tasks and to schedule those tasks in the context of a UK company – Locktel Limited.

Nowadays, Locktel Ltd. has four project managers who are responsible, among other duties, to allocate Locktel's engineers to tasks and schedule those tasks. The importance of this project lies in the fact that those allocations are performed manually. In spite of having a program – BigChange – to write down their final schedules, that later on are sent to the engineers, the used decision support system is not able to automatically solve the project managers' problem.

Similar problems have been described in the literature as the Technicians and Tasks Schedule Problem (TTSP) (Cordeau et al., 2010), and studied as being a specific case of Assignment Problems (AP) (Kuhn, 1955). TTSPs differ in a way from the problem presented in this master final work. The skills acquired from teams in a TTSP are just the sum of the skills of its members, while in this project each member must be assigned to a position in the team that allows its skills to be maximized. Nevertheless, it is undeniable that both problems have similarities and a good knowledge of one will help to understand the other.

The solution proposed corresponds to the development of an Excel program to assist project managers in their work. The program will be specifically designed for one manager. The Fibre to the Home's (FTTH) manager, being possible to adapt it to all Locktel's managers in the future. Important information regarding engineers and tasks will be exported from BigChange and imported to an Excel file. After that, target tasks will be scheduled and engineers will be assigned, by running the models. At the end, the generated information will be sent back to BigChange.

This Master's Final Work (MFW) is divided into five chapters. In the second, a literature contextualization is carried out. The third chapter is focused on the case study.

A results analysis and presentation of some enhancements is done in the fourth chapter. Lastly, some conclusions are drawn in the fifth and last chapter.

2. Literature Contextualization

The problem studied in this project consists in assigning engineers to tasks and in scheduling those tasks. For that reason, the problem is considered to be an Assignment Problem (AP) with side constraints. Firstly, in this chapter, AP problems will be introduced (2.1). Then, problems similar with the one studied in this project will be presented (2.2) as well as some differences between them.

2.1. Assignment Problems

The importance of an Assignment Problem (AP) lies in the fact that not just complex combinatorial optimisation problems can be relaxed or sub-divided and written as an AP, but also from its simplicity. Complex problems such as personnel rostering, facility location, maintenance scheduling, manpower and fleet planning, among others illustrate this statement (Abara, 1989; Mazzola & Neebe, 1986; Meneghetti & De Zan, 2016).

An AP, in its classic definition, occurs when a certain number of resources – in this case employees – need to be assigned to the same number of tasks – not every resource is necessarily legible to complete every task – and the aim is to maximize the total utility (Kuhn, 1955). The importance of AP is unquestionable. Therefore, since Kuhn (1955), numerous were the authors that generalized this problem by adding it side constraints. Side constraints are additional constraints to the classic model which diminish the gap between the modulated problem and the reality that it is trying to represent.

In the literature, side constraints vary from turning the problem to an unbalanced one, where there are more tasks than resources (Rabbani et al., 2019), to seniority and job priority, where senior employees' assignments are prioritised as well as the conclusion of top priority jobs (Caron et al., 2019). More examples can be seen in Aboudi & Nemhauser (1991), Kumar (2006), or Pathan & Shrivastava (2021), for instance.

The scheduling part of the problem, in other words, the need to schedule the tasks during a time period, is also important. An example of a Schedule Problem (SP) may be seen in Klingman & Ross (1973).

One particular AP with side constraints linked with a SP – Technicians and Tasks Schedule Problem (TTSP) – will be further and more profoundly addressed given its relevance and similarity with the present project.

2.2. Technicians and Tasks Schedule Problem

The Technicians and Tasks Schedule Problem (TTSP) aims to finish a project within its due time. To achieve that, a feasible schedule where all intended project's tasks are successfully completed needs to be determined (Bellenguez-Morineau & Néron, 2007). The connection with AP comes from the need to allocate technicians to tasks, and with SP comes from the need to schedule those latter.

In AP, resources are either legible or illegible to complete certain tasks. That information can be represented by a (binary) qualification matrix. Each row header represents a resource, each column header a task and the interception between row i and column j is 1 if resource i is legible to conclude task j, and 0 otherwise (Kuhn, 1955).

On the contrary, in TTSP, resources are not directly legible or illegible to complete tasks. Instead, each resource possesses a set of skills, and each task requires a set of skills (not necessarily the same) to be completed. When a resource, or a group of resources, meets all task's requirements than it becomes legible to complete it. For that reason, in TTSP, resources are commonly entitled technicians since they are qualified employees with more than one skill. This problem also appears in the literature as the Multi-Skill Project Schedule Problem (MSPSP) (Bellenguez-Morineau, 2008; Bellenguez & Néron, 2005).

There are many TTSP variants discussed in the literature. These vary from problems in which tasks are fixed in time and need to be assigned to technicians with predetermined working times, where the aim is to minimise the overall cost of personnel (Krishnamoorthy et al., 2012; Smet et al., 2014), to problems with flexible tasks' conclusion ranges where the main purpose is to minimise personnel travels or overall project time (Cordeau et al., 2010; Meneghetti & De Zan, 2016). Methodologies diversifying from branch-and-bound methods in the late 2000s (Bellenguez-Morineau & Néron, 2007) to hybrid constraint programming optimisation models solved by large neighbourhood search algorithms in more recent works (Meneghetti & De Zan, 2016) are proposed for the TTSP.

The problem addressed in this project may be addressed as a TTSP. The main difference lies in the objective function. This particular case aims for the maximisation of the skills of the chosen engineers in specific situations. Thus, instead of teams being assigned to tasks, engineers are assigned to tasks' positions. Since the problem has this particular objective function, the methodology will reside in developing two linked linear programming models (one mixed and one binary), as will be next addressed.

3. CASE STUDY

The telecommunication case study is detailed in this chapter. It is divided into four sections. Firstly, a general overview of the company – Locktel Limited – will be provided to contextualize the project (3.1). In the second section (3.2) the problem will be thoroughly described alongside with illustrative tables. The theoretical constructions gain life in the third section (3.3) where the two developed models are presented. To conclude, in the last section (3.4) some of the most important applications are exhibited and carefully explained.

3.1. Company Framework

"Locktel Ltd. is a leading telecommunications contractor based in Livingston, Scotland. With operations across the UK"[1]. Locktel is a "nationwide installer of fibre infrastructure for both homes and business" [1], providing all the materials as well as the labour force required for all stages of the process – from consultation to installation.

Locktel's Mission is to provide quality, safety, and innovative services to their clients. The Company's Vision is focused on the future of telecommunications and on what it can provide and inspire on communities. Values wise, the company upholds respect for the people using teamwork as the guidance to promote commitment, quality, and integrity.

Locktel's services vary among "Fibre to the Cabinet (FTTC) / Fibre to the Premisses (FTTP) builds, Data Network Installations, Closed-circuit Television (CCTV) & Access Control Installations, 5G Rollouts, Project Management, and Fibre to the Home (FTTH) / Business Installations"[1]. This project will focus on the latter service – FFTH / Business Installations.

[1] Available at Locktel Ltd. Website: https://locktelltd.co.uk (accessed on October 1st, 2021)

The specific service choice was mainly based on two aspects. Firstly, to accomplish installations in both homes and business there are prior notices that need to be done by Locktel and approved by the client. Consequently, the gap between knowing that an installation needs to be done and carrying it out is larger than in other telecommunications' services. That allows a more careful preparation. Secondly, "FTTH / Business Installations" are normally extensive and long-term projects, extending over months or even years. For this reason, when one installation arises, several tasks with a certain order of accomplishment need to be carried out.

Apart from unforeseen events, the above mentioned order of tasks grants project managers the possibility to plan weeks in advance. This planning process does not happen with such a wider window as it could since the allocation process is done by hand. However, this project aims to help project managers in that regard.

3.2. Problem description

The FTTH / Business Installations TTSP, which this project intends to solve, is detailed in this section. Locktel divides its project managers by cities. Therefore, when Locktel receives a new project from a certain city, the respective project manager has the responsibility to create, jointly with the administrative staff, the necessary tasks to accomplish that project. For that reason, Locktel has a continuously inflow of unscheduled tasks, each taking no longer than nine hours to complete, sourced from new projects. Each task is defined with a location, a description, a due time, an estimated duration, and a job type.

Depending on the job type, each associated task has its own skill requirements, as exemplified in Table I. In this table each column header represents a skill, while each row header represents a task. The interception between columns and rows shows the minimum number of engineers needed with that specific skill for that task.

Tasks \ Skills	Legal			Real		
	s_1	s_2	s_3	s_4	s_5	s_6
u_1	0	2	0	1	1	0
u_2	1	0	0	0	0	1

Table I. Tasks' Required Skills Matrix – Exemplification

Depending on the tasks' job type and also the project it belongs to, a task can have some immediate successor and/or predecessor tasks. In the example shown in Table II, each letter (A, B, C and D) stands for a different job type. The predecessors of each job type are depicted in the correspondent row (e.g. A precedes B in Table II, third row), while successors can be found in columns (e.g. D succeeds B and C , Table II, third and fourth columns).

Job Type \ Immediate Predecessor	A	B	C	D
A		0	0	0
B	1		0	0
C	0	0		0
D	0	1	1	

Table II. Immediate Predecessor Job Type Matrix – Exemplification

There are engineers with a known set of skills that can complete the mentioned tasks, and which work independently for Locktel. In other words, these engineers are self-contractors – and Locktel only has to pay them according with their working hours –, each one with a corresponding set of skills. Those skills are named as legal (s_1 to s_3 in Table III) or real (s_4 to s_6 in Table III).

Legal skills are certified and imposed by legislation to perform certain tasks (mainly courses, awareness courses and compulsory certifications), and have an expiration date. Each engineer may have it or not.

Real skills are the materialization of Locktel's evaluation in specific aspects of the tasks. Each engineer can be evaluated from 0 to 5 depending on the degree of expertise (5 being the highest/best evaluation). This evaluation is only recognized internally by Locktel. Real skills can differ between skills for leader or mate positions. This real skills' orientation will be thoroughly addressed latter.

Engineers \ Skills	Legal			Real		
	s_1	s_2	s_3	s_4	s_5	s_6
e_1	1	1	1	5	4	1
e_2	0	1	1	5	4	0
e_3	0	1	0	3	5	0
e_4	1	0	0	0	0	5

Table III. Engineers' Skills Matrix – Exemplification

To schedule tasks is a responsibility of Locktel's project managers. It is also intended for them to group engineers in teams, when necessary, and then to allocate teams or individual engineers to tasks. Although Locktel uses BigChange – a software to manage salaries, employees, and other activities – all this process of allocation is currently done by hand, as already stated. BigChange has an integrated allocation system within the program, however it is not specifically designed for Locktel's needs. Although the system recommends engineers, it is not able to automatically assign them satisfying all required restrictions. Project managers' aim is to complete all tasks before their due time by assigning them the more capable and skilled (within the required skills) engineer or set of engineers.

Some tasks require two engineers. That necessity can only be seen in legal skills (see for example s_2 of u_1 in Table I). In such cases, the task also requires at least one real skill for a leader and at least one for a mate. Leadership skills are specifically required for the first assigned engineer who is responsible for the task. Mate-skills are needed for the second engineer to possess. To fully address those cases, each task in need of two engineers will be divided in two sub-tasks. The first sub-task will require one engineer, the leader (exemplified by u_{1L} in Table IV), while the second will required one other engineer, the mate (illustrated by u_{1M} in Table IV). In the example, s_4 represents a leadership skill, whereas s_5 a mate-skill (compare with matrix prior to modifications, Table I), as it is possible to see by comparing Table I and Table IV (u_2 remains the same in both tables since it only requires one engineer to be completed).

Tasks \ Skills	Legal			Real		
	s_1	s_2	s_3	s_4	s_5	s_6
u_{1L}	0	1	0	1	0	0
u_{1M}	0	1	0	0	1	0
u_2	1	0	0	0	0	1

Table IV. Modified Tasks' Required Skills Matrix Illustration

A solution to this problem was tried to be accomplished by developing a mixed integer model. However, with a single model it would not be possible to separately maximise both the leaders' and the mates' required skills. Since leaders and mates are required for different positions in the task.

In conclusion, and in order to pursue the maximisation of the required skills in all situations, the defined problem was solved through two sub-problems. The first sub-problem (presented in 3.3.1) schedules all tasks and assigns them an engineer, the leader, while the second (presented in 3.3.2) assigns a second engineer, the mate, to the required positions, taking as input the solution of the first sub-problem.

3.3. Mathematical Models

This section presents a mixed integer linear model and a binary linear model to define and to solve the above described problem. This section is divided in two sub-sections. The first (3.3.1) aims to address the tasks that need a leader or only one engineer, scheduling the tasks and assigning them engineers, whilst the second one (3.3.2) looks at the assignment of mates. This latter sub-section will use the output of the first model as its own input, as already stated.

3.3.1. Leaders' Model

The purpose of this first sub-section is to present the mixed integer linear model and to state all the information needed to generate an optimal solution for the Leaders' part of the problem, respecting their working hours and all the above mentioned restrictions. In other words, the model will allocate, to the identified tasks, the more capable, suitable, and available engineers – making them the leaders. Although Locktel has a special invoicing system for overtime, this will not be considered in this model. All the definitions and notation can be found underneath (3.3.1.1) as well as decision variables (3.3.1.2), objective function (3.3.1.3) and restrictions (3.3.1.4).

3.3.1.1.Definitions and Notation

$E = \{1, \dots, |E|\}$ – set of engineers.

$S = \{1, \dots, |S|\}$ – set of skills.

$U = \{1, \dots, |U|\}$ – set of unscheduled tasks, also referred to as tasks, for simplification.

$U^l = \{1, \dots, |U^l|\}$ – set of unscheduled tasks in need of a leader, $U^l \subseteq U$.

$\nu_s^e \in \{0, 1, \dots, 5\}$ – expertise of engineer e, $e \in E$, in skill s, $s \in S$.

$\upsilon_s^u \in \{0, 1\}$ – equal 1 if task u, $u \in U$, requires the assigned engineer to possess skill s and 0 otherwise, $s \in S$.

λ_u – due time, in date-time serial number, of task u, $u \in U$, e.g. $\lambda_1 = 44371{,}375$ represents that task 1 must be completed before 9 AM of June 24, 2021. The integer part of the date-time serial number – 44371 – refers to the day, i.e. 24/06/2021 (44371 days after the base day). The decimal part – 0.375 – represents the time i.e. 09:00:00 (37.5% of the day).

$\delta_u \in [0, 0.375]$ – duration, in time serial number, of task $\mathrm{u}, u \in U$. In other words, how long it takes to successfully complete task u, $u \in U$. Note that each task takes no longer than nine hours (37.5% of the day) to complete.

Π_u – set of immediate predecessor tasks of task $u, u \in U$. In other words, set of tasks that are directly linked with task u and that must be finished before task u starts, $u \in U$.

D – set of days, in date serial number.

E^d – set of engineers available at the day d, $d \in D, E^d \subseteq E$.

$\mathcal{B} \in [0,1]$ – value representing the beginning, in time serial number, of a working day.

$\mathcal{C} \in [0,1]$ – value representing the closing, in time serial number, of a working day.

$\mathcal{M} = max(d) + 1$, $d \in D$.

3.3.1.2. Decision variables

$x_{eud} = \begin{cases} 1, \text{if engineer } e \text{ is assigned to task } u \text{ at day } d \\ 0, \text{otherwise} \end{cases}, d \in D, \mathrm{e} \in E^d, u \in U^l.$

$b_{uu'} = \begin{cases} 1, \text{if task } u \text{ is completed before task } u' \text{ starts} \\ 0, \text{otherwise} \end{cases}, u, u' \in U^l, \text{with } u < u'.$

z_u – starting time, in date-time serial number, of task u, $u \in U^l$.

3.3.1.3. Objective Function

$$Max\left\{\sum_{s \in S} \sum_{u \in U^l} \sum_{d \in D} \sum_{e \in E^d} v_s^u v_s^e x_{eud}\right\} \quad (1)$$

The objective function (1) maximizes the total skills of the chosen engineers according to the skills required by the tasks.

3.3.1.4. Restrictions

$$
\begin{cases}
z_u + \delta_u \leq \lambda_u, \ \forall u \in U^l & (2) \\
\sum_{d \in D} \sum_{e \in E^d} x_{eud} = 1, \quad \forall u \in U^l & (3) \\
v_s^u x_{eud} \leq v_s^e x_{eud}, \quad \forall d \in D, \forall e \in E^d, \ \forall u \in U^l, \ \forall s \in S & (4) \\
z_u + \delta_u - z_{u'} \leq 0, \quad \forall u \in \Pi_{u'}, \forall u' \in U^l & (5) \\
\mathcal{B} + \sum_{d \in D} \sum_{e \in E^d} d x_{eud} - z_u \leq 0, \quad \forall u \in U^l & (6) \\
\mathcal{C} + \sum_{d \in D} \sum_{e \in E^d} d x_{eud} - (z_u + \delta_u) \geq 0, \quad \forall u \in U^l & (7) \\
z_u + \delta_u \leq z_{u'} + \mathcal{M}(2 - x_{eud} - x_{eu'd}) + \mathcal{M}(1 - b_{uu'}), & \\
\qquad \forall u, u' \in U^l : u \leq u', \ \forall d \in D, \ \forall e \in E^d & (8) \\
z_{u'} + \delta_{u'} \leq z_u + \mathcal{M}(2 - x_{eud} - x_{eu'd}) + \mathcal{M} b_{uu'}, & \\
\qquad \forall u, u' \in U^l : u \leq u', \ \forall d \in D, \ \forall e \in E^d & (9) \\
x_{eud} \in \{0,1\}, \qquad \forall d \in D, \forall e \in E^d, \forall u \in U^l & (10) \\
b_{uu'} \in \{0,1\}, \qquad \forall u, u' \in U^l : u \leq u' & (11) \\
z_u \geq 0, \qquad \forall u \in U^l & (12)
\end{cases}
$$

Inequalities (2) impose tasks to finish before their due time while inequalities (3) assign tasks to only one engineer at one day. Constraints (4) ensure that tasks are only assigned to engineers that fulfil all tasks' required skills. Inequalities (5) enforce the immediate precedence constraints. In other words, enforce tasks with immediate predecessors to start only after their immediate predecessors are completed.

Inequalities (6) and (7) state that tasks assigned at a specific day must start, at least, at the day's first moment ($d + \mathcal{B}$) and finish until the day's last moment ($d + \mathcal{C}$). Constraints (8) and (9) guarantee that tasks assigned to the same engineer at the same day do not occur at the same time. Thus, if $x_{eud} + x_{eu'd} = 2 \Longrightarrow z_u + \delta_u \leq z_{u'}$ (if $b_{uu'} = 1$) or $z_{u'} + \delta_{u'} \leq z_u$ (if $b_{uu'} = 0$).

Constraints (10) to (12) define the domains of the decision variables.

3.3.2. Mates' Model

The binary linear model developed to define the mates' part of the problem will be presented in this sub-section. As in the leaders' model, all the definitions and notation can be found underneath (3.3.2.1), as well as the decision variables (3.3.2.2), the objective function (3.3.2.3) and restrictions (3.3.2.4).

This second model will use the output of the leaders' model (available at 3.3.1) as input, as previously stated. Moreover, all tasks demanding only one engineer as well as their already assigned engineers will not be considered. Additionally, engineers already assigned to two-engineer tasks at specific days are considered as leaders on those days, and the ones not yet assigned at specific days are potential mates on those days. It can happen that one engineer is assigned to a two-engineer task at day 1, therefore being leader at day 1, and be a potential mate on the remaining days as he was not yet assigned to any task in those days.

Note that, since the models are connected, all definitions and notation previously stated (in 3.3.1.1) still apply.

3.3.2.1.Definitions and Notation

$L^d = \{1, \dots, |L^d|\}$ – set of leaders assigned at day d, $d \in D$, that is $e \in L^d$ if $e \in E^d \wedge \sum_{u \in U^l} x_{eud} = 1, \forall d \in D$.

$M^d = \{1, \dots, |M^d|\}$ – set of mates available at day d, $d \in D$, i. e. $e \in M^d$ if $e \in E^d \wedge \sum_{u \in U^l} x_{eud} = 0, \forall d \in D$.

$U^m = \{1, \dots, |U^m|\}$ – set of tasks in need of a mate, $U^m \subseteq U$.

$$x_{lud} = \begin{cases} 1, \text{if leader } l \text{ is assigned to task } u \text{ at day } d \\ 0, \text{otherwise} \end{cases}, d \in D, l \in L^d, u \in U^m.$$

3.3.2.2.Decision variables

$$y_{mld} = \begin{cases} 1, \text{ if mate } m \text{ works with leader } l \text{ at day } d \\ 0, \text{ otherwise} \end{cases}, \forall d \in D,\ \forall m \in M^d,\ \forall l \in L^d.$$

3.3.2.3. Objective Function

$$Max \left\{ \sum_{s \in S} \sum_{d \in D} \sum_{m \in M^d} \sum_{u \in U^m} \sum_{l \in L^d} v_s^u v_s^m y_{mld} \right\} \tag{13}$$

The objective function (13) maximizes the total skills of the chosen mates in the skills required by the tasks.

3.3.2.4. Restrictions

$$\begin{cases} \sum_{m \in M^d} y_{mld} = 1, \quad \forall d \in D, \forall l \in L^d & (14) \\ \sum_{l \in L^d} y_{mld} \le 1, \quad \forall d \in D, \forall m \in M^d & (15) \\ v_s^u x_{lud} \le \sum_{m \in M^d} v_s^m y_{mld} \quad \forall d \in D, \forall l \in L^d, \forall s \in S, \ \forall u \in U^m & (16) \\ y_{mld} \in \{0,1\}, \quad \forall d \in D, \forall m \in M^d, \forall l \in L^d & (17) \end{cases}$$

Equalities (14) state that each leader must have one mate, per day, working with him. Inequalities (15) guarantee that mates can only work with at most one leader per day. Inequalities (16) ensure that mates are only assigned to leaders if mates fulfil all tasks' required skills, and constraints (17) define the domain of the decision variables.

3.4.Empirical Applications

The empirical applications of this MFW were achieved with the development of a program in Excel (see Appendix I). The program was written using Visual Basic for Applications (VBA) language. In total, 32 macros were developed within eight modules, which can be roughly divided in three core parts.

The first part (3.4.1) is focused on the import of data from BigChange to the Excel program and its validation. The second and most essential (3.4.2) is centred on sequentially solving the above presented models (3.3.1 and 3.3.2) using OpenSolver. The last part (3.4.3) is dedicated to the process of exporting data from the Excel program back to BigChange.

3.4.1. Importing Data

Firstly, the desired data must be exported from BigChange and saved in the same folder as the Excel program. This data is related with engineers – resources, in BigChange and Locktel language – and tasks – jobs. When the user opens the Excel program is directed to its main sheet (Figure 1). Here the user is able to fully access all program spectrum via buttons.

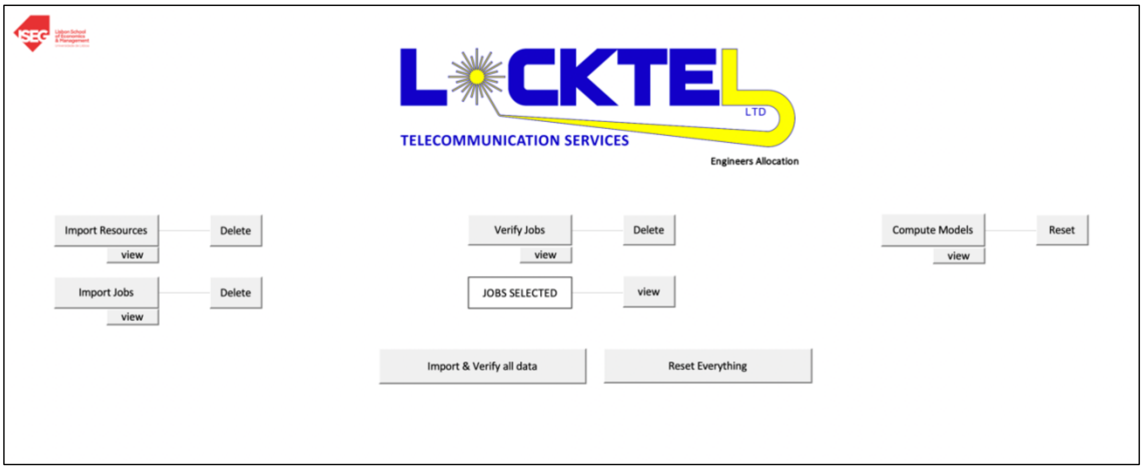

Figure 1 – Program's Main Sheet

To import and verify the data the user can either do it separately – pressing "Import Resources", "Import Jobs" and "Verify Jobs" buttons – or all at once – with the "Import & Verify all data" button.

When the user presses the "Import Resources" button, the program will search – in the same driver as the Excel program is located – for a document named "Resources Export", with the help of the "ActiveSheet.QueryTables.Add" method. The user will be asked to identify – within an "InputBox" – the type of resource is currently working on (Figure 2). The predefined answer is "Engineers Scotland" because these empirical applications were focused on a Scottish project. From this moment onwards, engineers from other countries or even other Scottish resources besides engineers, like managers or administrative staff, will no longer be considered by the program (for a clearer understanding of the process behind this button see Figure 11, Appendix 2).

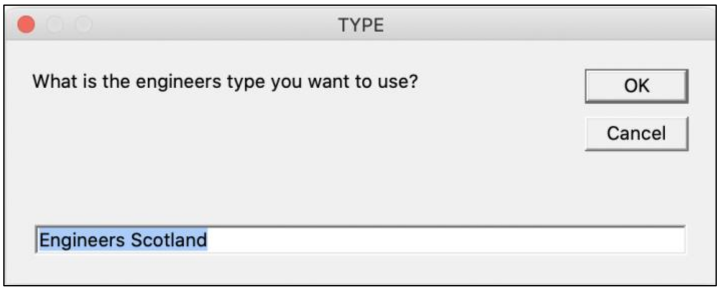

Figure 2 – Resources' Type

Once all data is imported, the cells alongside the button are filled in green and a message appears to notify the user (Figure 3).

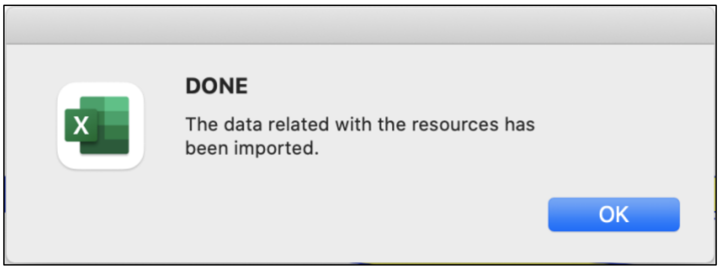

Figure 3 – Imported Resources Confirmation Message

A similar occurrence happens when pressing the “Import Jobs” button. However, this time the program will search for a document named “Jobs Export” (see Figure 12, Appendix 2). An error message as the one in Figure 4 indicates that the user tried to import data for the second time without clearing data between imports.

Figure 4 – Error Message

The above message appears if the user tries to repeat the import of data related with the jobs. However, a similar message appears in other cases. To avoid those messages, the user must delete old information before trying to import new one. For that, the nearest “delete” or “reset” button (Figure 1) should be pressed. After, the user will be asked to confirm the decision (Figure 5).

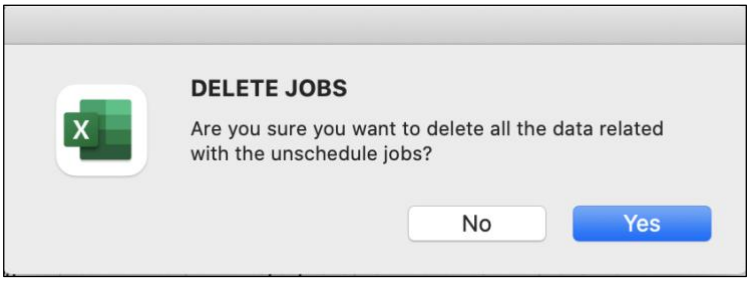

Figure 5 – Delete Jobs Decision Confirmation Message

When pressing the “Verify Jobs” button the user is asked to name the project or projects the looked-for jobs are related to. Other project-related jobs will be disregarded by the program. Whilst the program filters the desired jobs, also verifies its information, and creates a “select” button for each job (see Figure 13, Appendix 2). The “select” button will be further explained in 3.4.2.

The global “Import & Verify all data” button, as already stated, works as if all previously mentioned buttons were automatically pressed (see Figure 14, Appendix 2). Then, every nearby cell is shaded in green, which allows users to easily know that all data was already successfully imported and verified (Figure 6).

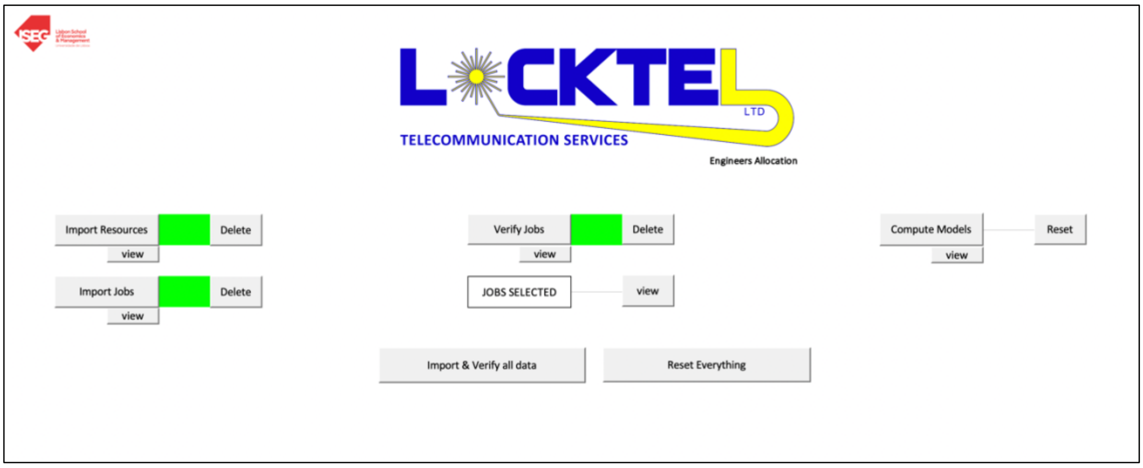

Figure 6 – Program’s Main Sheet After Importation & Validation Processes

3.4.2. *Solving Models*

The first indispensable action required from the user is to select the desired jobs. The selected jobs are considered as input in both models, being divided first into sub-jobs, whenever requiring a leader and a mate (see Figure 15, Appendix 2). To select the required jobs, the user must view all the already verified jobs, by pressing the nearest “view” button from the “Verify jobs” button. After, the user must select the desired job among the verified ones by pressing the “select” button.

At this point, the program possesses all the information needed to solve the models. In other words, all the information required to schedule the chosen jobs and to assign them the most capable engineers. This is, in fact, the output whenever the user presses the “Compute Models” button (see Figure 16, Appendix 2). The program gathers all the information from all necessary sheets and computes everything in a new sheet called “MODEL”.

First and foremost, the leaders' model will be defined (see Figure 17, Appendix 2). The rows' and columns' headers that will define the limits of the decision variables are expressed in the "MODEL" sheet. Then, the objective function and all restrictions are written using the "ActiveCell.FormulaR1C1" property. Afterwards, all cells with restrictions, objective function or decision variables are transposed into OpenSolver program taking advantage of the "SolverAdd" function. Then, the solution is determined by the OpenSolver program. Once it finds an optimal solution, the decision variables are saved in a new sheet. This new sheet can be defined as the output sheet of the first model as well as the input of the second.

For the second model, like for the former, the decision variables' boundaries will be written down, followed by the objective function and all restrictions. Afterwards, all relevant cells will be transposed once again to the OpenSolver and, subsequently, the program will run. Immediately after an optimal solution is generated, the output of this second model is added to the output sheet of the first (see Figure 18, Appendix 2), to complete the final solution.

3.4.3. Exporting Data

Exporting data is the last part of the program. After the models have run, the generated solutions are saved into a sheet, so that the user can verify them. All the information presented in this sheet is organized in such a way that BigChange is able to automatically import it.

Note that project managers may want to adapt the solution. Among other reasons, project managers may want to assign neighbour engineers to the same task – potentially swapping skills' quality by favouring engineers' comfort and reducing travel times. Additionally, an engineer can be in training, and therefore can be assigned as the third engineer in some tasks (not considered in this project). Some of these issues can be further implemented in the program and will be addressed in the next chapter (see 4.2). But ultimately, project managers will always need to check the computed solutions as they may have preferences and non-quantifiable knowledge that are very challenging to include in the models.

After the solution is checked by the user, the information needs to be exported back to BigChange. The user must then press the "Export" button, available in the sheet which has the solution (Figure 7). As soon as the button is pressed, the solution sheet will be exported to an independent Excel file named "BigChange" using the "Workbooks.Add" method. When importing the file to BigChange all chosen tasks will be scheduled and intended engineers assigned. All the relevant information – regarding the tasks and the engineers – within BigChange will be used, e.g. engineers' vehicles, tasks' location, and others. All new information provided by the solution generated by the program will be added, namely assigned engineers, tasks' date, and estimated time of conclusion.

Job reference	Export	Job type	Resource name	Job details	Planned start time
jobref1		C	e12		14/09/2021 08:00
jobref2		N	e32 e13		14/09/2021 16:40
jobref3		N	e32 e13		14/09/2021 16:20
jobref4		N	e32 e13		14/09/2021 16:00
jobref5		N	e32 e13		14/09/2021 15:40
jobref6		N	e32 e13		14/09/2021 09:00
jobref7		N	e32 e13		14/09/2021 08:40
jobref8		N	e32 e13		14/09/2021 08:20
jobref9		N	e32 e13		14/09/2021 08:00
jobref10		G	e15		14/09/2021 16:30
jobref11		G	e15		14/09/2021 08:30
jobref12		G	e15		14/09/2021 08:00
jobref13		D	e5 e14		14/09/2021 08:00
jobref14		M	e19		14/09/2021 15:00
jobref15		M	e6		14/09/2021 08:00
jobref16		M	e2		14/09/2021 08:00
jobref17		K	e34 e21		14/09/2021 15:00
jobref18		K	e34 e21		14/09/2021 10:00
jobref19		I	e11		14/09/2021 08:00
jobref20		C	e38		14/09/2021 08:00
jobref21		D	e24 e47		14/09/2021 08:00
jobref22		K	e34 e21		14/09/2021 08:00
jobref23		C	e23		14/09/2021 08:00

Figure 7 – Export Solution Sheet

4. Achievements' Analysis

The process of evaluation is important in every project. The analysis performed in this chapter is focused on comparing the models' solutions with the ones designed by project managers. For that purpose, five specific scenarios are identified in the first section (4.1), whereas some potential enhancements to the model are discussed in the second (4.2).

4.1. Scenarios

Five one-day scenarios were extracted from one real working week. Firstly, one scenario will be presented and analysed in detail. Then, a summary table accounting the entire week data will be displayed jointly with some comments and conclusions.

The first scenario concerns the scheduling of 13 independent tasks in one day. Therefore, there are no precedence constraints. Five of those 13 tasks require more than one engineer. There are 46 available engineers, and a total of 90 skills. In total, in the first model, there are 689 decision variables that need to be determined and optimised (18). In the first model, 9 engineers were assigned, two of those became leaders in the second model. Therefore, there are 37 potential mates. So, in the second model there are 74 decision variables (19).

Decision Variables

$$1^{st}\text{ model} \rightarrow x_{eud}\ ;\ b_{uu'}\ ;\ z_u: (46 \times 13 \times 1) + (\tfrac{12 \times 13}{2}) + 13 = 689 \tag{18}$$

$$2^{nd}\text{ model} \rightarrow y_{mld}: 37 \times 2 \times 1 = 74 \tag{19}$$

In Table V the FTTH project manager's choice (columns 4 and 5) as well as the program's solution (columns 6 and 7) are shown. For simplification when addressing solutions, project manager's solution will be named A and the program's solution will be B.

Tasks	Tasks' Duration	Job Type	Project Managers' Solution (A)		Program's Solution (B)	
			Engineers	Start Time	Engineers	Start Time
u_1	1h	D	e_{17} e_{13}	13/09/2021 08:00	e_{47} e_{13}	13/09/2021 16:00
u_2	1h	D	e_{17} e_{13}	13/09/2021 13:00	e_{47} e_{13}	13/09/2021 15:00
u_3	1h	D	e_{17} e_{13}	13/09/2021 16:00	e_{47} e_{13}	13/09/2021 09:00
u_4	1h	M	e_{19}	13/09/2021 10:00	e_{22}	13/09/2021 08:00
u_5	1h	M	e_{19}	13/09/2021 11:00	e_{19}	13/09/2021 16:00
u_6	1h	M	e_{19}	13/09/2021 12:00	e_{19}	13/09/2021 08:00
u_7	1h	D	e_5 e_{14}	13/09/2021 08:00	e_{47} e_{13}	13/09/2021 08:00
u_8	1h	D	e_5 e_{14}	13/09/2021 10:00	e_3 e_{14}	13/09/2021 08:00
u_9	1h	C	e_{23}	13/09/2021 08:00	e_6	13/09/2021 08:00
u_{10}	1h	C	e_{23}	13/09/2021 14:00	e_{54}	13/09/2021 08:00
u_{11}	9h	I	e_{11}	13/09/2021 08:00	e_{11}	13/09/2021 08:00
u_{12}	9h	I	e_{11}	13/09/2021 08:00	e_{36}	13/09/2021 08:00
u_{13}	6h	J	e_{26}	13/09/2021 08:00	e_{23}	13/09/2021 08:00

Table V. First Scenario, Solution A and B

Figure 19 and Figure 20, in Appendix 3, represent the schedules identified in Table V, for solution A and B, respectively. As it is possible to observe, in solution B, neither the minimisation of number of engineers nor the finish time of the last task were considered in the objective function, otherwise, the schedule would have a substantially different aspect. If the minimization of engineers was pondered, the number of chosen engineers would be diminished and engineers' working hours would tend to be higher, while if the finish time of tasks was considered, engineers' working hours would tend to be just the length of one task.

Table VI expresses some metrics that allow comparation between the previous presented solutions (OFV stands for Objective function's value).

	Solution A	Solution B
Engineers Assigned	8	11
$\frac{\text{Nr. Tasks}}{\text{Nr. Engineers}}$	1.63	1.18
Leaders' OFV (max skills)	89	115
$\frac{\text{Leaders' skills}}{\text{Nr. tasks}}$	6.85	8.85
Mates' OFV (max skills)	35	35
$\frac{\text{Mates' skills}}{\text{Nr. tasks}}$	7	7
Working Hours – Min	2h	1h
Working Hours – Average	5h40	3h46
Working Hours – Max	18h	9h
Last finished task time	17:00	17:00
Discrepancy – equation (20)		61.29%

Table VI. First Scenario, Manager-Program Solutions Comparison

As it may be seen in Table VI, solution B has higher objective function value (115) than A (89), in the leaders' part of the problem. Thus, leaders in B, according with the worked data, should be more capable to successfully complete their tasks than leaders in A. That is no surprise since the program will find an optimal solution, if there is one, therefore choosing engineers based on their expertise.

It is important to highlight that locations regarding tasks and engineers' houses were not considered. In other words, the gap between the values of the two objective functions could be due to a reduction in the number or distance of travels made by the assigned engineers in solution A. That increase in engineers' well-being was not possible to quantify under the available information, neither from BigChange, nor the managers. Nevertheless, it is a concern that project managers have into account, and if included, will improve the generated solutions from the engineers' point of view.

Another relevant aspect of solution A is that it does not represent a feasible solution. This is possible to understand by carefully analysing Table V, especially task u_{11} and u_{12}

assigned engineer (e_{11}) and start time (08:00). Both tasks are assigned to the same engineer at the same time. Moreover, each one of those tasks take approximately 9 hours to be successfully completed. That justifies the value of 18h for the maximum working hours in solution A (Table VI). This occurrence is impossible under the model's restrictions. However, a project manager can accurately know both tasks and evaluate them as achievable in the working time period by the assigned engineer. If indeed both tasks are possible to complete in 9h, it would be interesting to consider both tasks as only one, that would allow the program to work with the same information as the project managers, tasks' duration wise.

Analysing the metrics in Table VI, solution B seems better than solution A. In spite both solutions finish tasks at the same time (17:00), solution B uses more capable engineers than solution A, on average. But there are grey areas. In fact, solution B has a better working hour range, but due to the usage of more engineers.

Despite having solution B with more capable engineers, the program was created to help project managers in their work. And even with the knowledge of solution B, FTTH project manager can still prefer solution A. However, the project manager would need to change less than two thirds of solution B to end up with solution A, since the discrepancy between solution B and A is 61.29% (Table VI, last column). Thus, the program helps its work as the FTTH project manager would not need to create solution A from scratch.

Both solutions have 18 spaces filled with engineers (number of tasks plus number of tasks in need of a mate) and 13 filled with dates (number of tasks). The discrepancy (20) between solution B and A is calculated dividing the number spaces filled with different information in the solutions by the sum of all spaces.

$$Discrepancy = \frac{\sum Engineers\ assigned\ to\ different\ tasks + \sum Tasks\ scheduled\ at\ different\ times}{2 \times Number\ of\ tasks + Number\ of\ tasks\ in\ need\ of\ mate} \quad (20)$$

To assess the solutions obtained from the models in more heterogeneous environments, the process applied to the first scenario was repeated for an entire week, five days. Table VII shows results for that working week, where the first already presented scenario is part of.

	Solution A			Solution B		
	Min	Average	Max	Min	Average	Max
Engineers Assigned	8	15	18	11	16	19
$\frac{\text{Nr. Tasks}}{\text{Nr. Engineers}}$	1.28	2.00	3.15	1.18	1.74	2.16
$\frac{\text{Leaders' skills}}{\text{Nr. tasks}}$	4.90	6.03	6.85	7.29	7.68	8.85
$\frac{\text{Mates' skills}}{\text{Nr. tasks}}$	2.56	4.56	7.00	2.56	4.71	7.00
Working Hours	00h30	04h22	18h00	00h30	03h53	09h00
Discrepancy				54.24%	63.94%	70.41%

Table VII. One Week, Manager-Program Solutions Comparison

First, it is relevant to highlight that solution B acquired a higher Leaders' OFV per task in every scenario, since solution B minimum (7.29) is bigger than solution A maximum (6.85) (Table VII). Therefore, as in the first scenario (Table VI), solution B is the one that chooses the more capable engineers to a leader position. Solution B has also a higher Mates' OFV in some cases, however, the difference is not significant.

Secondly, the grey area between engineers' having a better working schedule due to the usage of more engineers in solution B, remains. As it is possible to observe in Table VII, solution A uses fewer engineers than B (see "Engineers Assigned" line in Table VII), at the cost of the assigned engineers having to work more, on average (see "Working Hours" line in Table VII). This occurrence is not necessarily negative, but since the engineers are self-employers and earn per hours worked, that means solution A has a more unbalanced distribution of the engineers' wages than B. Moreover, once again we are faced with the solution A unreal maximum engineer' working hour (18h00). However, all tasks' durations are estimated and, once more, the knowledge of project managers is evident when understanding tasks and evaluating their real duration. Note that since the weight of each scenario is unknown, an average of the results was used and every scenario assumed as equal, weight wise.

The discrepancy between solution B and A can only be explained by two aspects. The first, solution B needs enhancements to recreate the project managers' choice more closely. The second, solution B is better than solution A, according with the defined objectives, and that is the reason why the solutions are different. It would be interesting to make a more careful analysis to the discrepancy between solutions to measure how much of its percentage is related with which of the above stated aspects.

Lastly, it was noticed that the first scenario (Table VI), the one that has less tasks, represents the maximum in Leaders' and Mates' skills per task in both solutions (Table VII). Therefore, two line charts were generated (Figure 8 and Figure 9) with information of the previously analysed week. It is possible to see the relation between the number of tasks that need to be scheduled in one day and the average qualification of the assigned engineers in that same day.

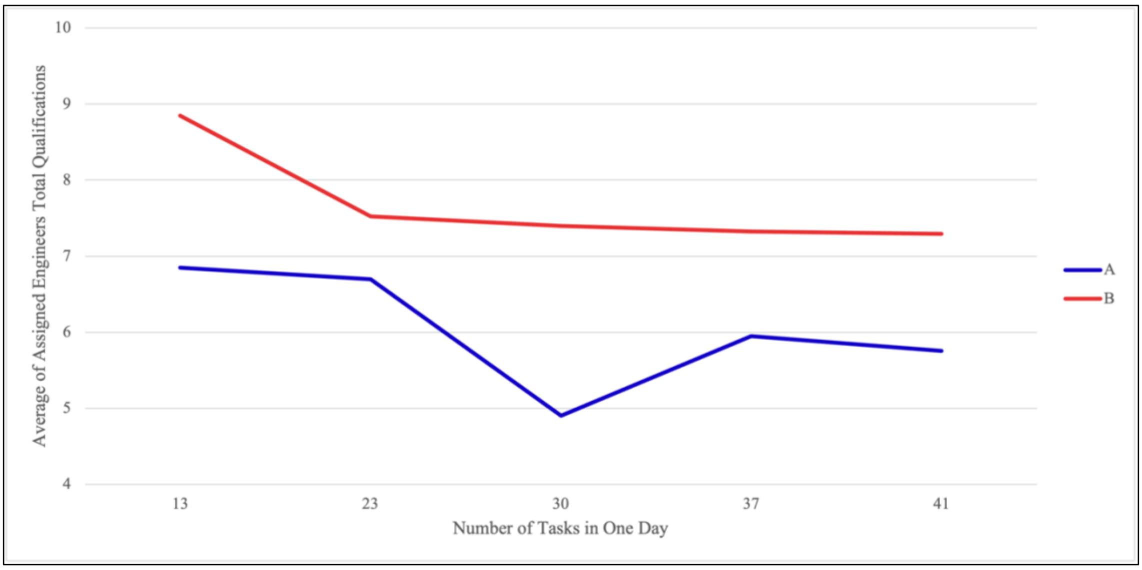

Figure 8 – Leaders' Objective Function Value per Task

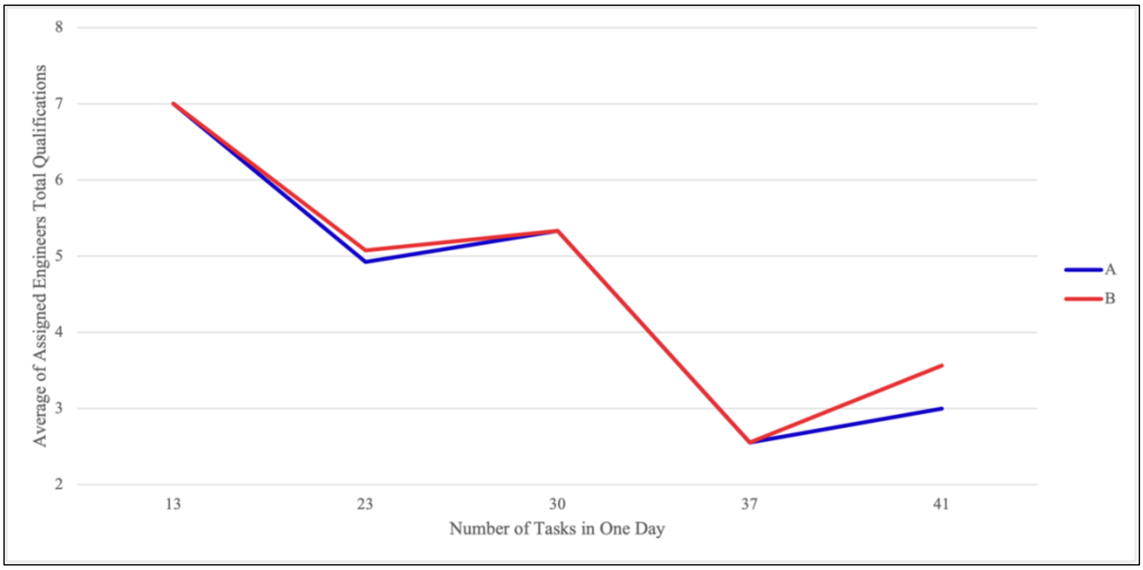

Figure 9 – Mates' Objective Function Value per task

In conclusion, the aim of this project was achieved, as project managers have access to easier ways to schedule tasks and assign engineers – qualifications and resources managing wise. Nevertheless, the need to implement modifications to recreate the project managers' way of thinking more accurately was felt, and some enhancements which point in that direction will be briefly addressed in the next section.

4.2. Enhancements

Some features are briefly discussed in this section in order for the proposed solution move closer to the project managers' way of thinking and accurately define this particular problem at any circumstance.

4.2.1. Priority Side Constraints

As already studied in the literature (Caron et al., 2019), side constraints related with tasks' priority can add value to the model when in the presence of different priority projects. This priority scale is independent from due times or other already stated parameters. Note that, since these priority constraints affect the schedule of tasks, – and not the assignment of engineers – these new constraints would only appear in the first model. The underneath proposed definitions follow the notation presented in 3.3.

Proposed Definitions:

- $P = \{1, \ldots, |P|\}$ – set of priority levels, the higher the more prioritised.
- ρ – priority level, $\rho \in P$.
- U^{ρ} – set of tasks with priority ρ, $\rho \in P$.

Decision Variable's example:

- e^{ρ} – ending time of the latest task with ρ, $\rho \in P$.

Restrictions' example:

- $e^{\rho} \geq z_u + \delta_u, \forall \rho \in P, \forall u \in U^{\rho}$ (21)
- $e^{\rho} < e^{\rho'}, \forall \rho \in P: \rho > \rho'$ (22)
- $e^{\rho} \geq 0, \forall \rho \in P$ (23)

Inequalities (21) ensure the ending times are well computed whilst constraints (22) guarantee that the set of tasks with higher priority level finish before other priority levels' set of tasks. Constraints (23) define the domains of e^{ρ} variables.

4.2.2. *Get Along Matrix*

A second improvement is related with teams, that is when two or more engineers are assigned to the same task. Available engineers are joined to perform tasks due to their correspondingly set of qualifications, notwithstanding of their compatibility to work together. If there was a “get along matrix” (see Table VIII), the model could consider the satisfaction of engineers when working together.

This matrix could be scaled from 0 – get along badly – to 5 – get along really well – and it could be the output of an anonymous query. The interception between rows and columns would represent how much does the engineer in the rows’ header – engineer i in Table VIII – would enjoy working with the engineer in the columns’ header – engineer j in Table VIII. For example, in Table VIII, engineer e_1 gets along badly with engineer e_3 (0 – second row, fourth column), and the vice versa also occurs (1 – fourth row, second column). It is also shown that engineer e_3 enjoys working with engineer e_4 (4 – fourth row, fifth column) and it is reciprocated (5 – fifth row, fourth column).

Engineer i \ Engineer j	e_1	e_2	e_3	e_4
e_1		3	0	3
e_2	3		2	3
e_3	1	2		4
e_4	3	3	5	

Table VIII. Get Along Matrix Exemplification

4.2.3. *Routing Extension*

Project managers have locations in consideration. Either the location of the unscheduled tasks or of the engineers. That allows managers to perform the assignments compromising assigned engineers’ qualifications in favour of keeping them in the same area during a day or even in an area nearby their houses at the end of the working day.

All specified improvements, regardless of its added value, had a reason not to be implemented in the presented project. Since nowadays the company does not practise a job priority scale, neither anything similar to a get along matrix, such improvements would need to be first created and updated in order to be introduced in the model.

5. CONCLUSION

The presented project consisted in the development and implementation of a Mixed Integer Linear Problem and a Binary Linear Problem to allocate multi-skilled engineers to multi-skill tasks and to schedule those tasks in the context of a UK company – Locktel Limited.

Similar problems have been described in the literature as Technicians and Tasks Schedule Problem (TTSP), based on an Assignment Problem mixed with a Schedule Problem. The presented problem differs from the TTSP as each member must be assigned to a position in the team that allows its skills to be maximized. Thus, two linked models were developed. The first model schedules all tasks and assigns one engineer per task, the most capable – leader. The second identifies which tasks, within the already scheduled tasks, need more engineers – mates – and assigns the most capable mate to each leader. This may be considered as bi-objective problem solved hierarchically.

Conclusions were drawn after the analysis done in chapter four. First and foremost, all proposed solutions were better than the ones done by the project manager, engineers' qualifications wise, confirming this project's added value. Moreover, project managers may be looking for schedules with the minimum number of engineers or even schedules with the earliest "last finished task time" and the objective function can be adapted to find those schedules. For that reason, the Excel program achieved the proposed goal, which was help project managers in their work.

Lastly, in regard of future developments, besides the highlighted enhancements, there are two more that stand out. The first one is to implement all possible objective functions in the leaders' model and ask for project managers to fill in with weights for each objective function. Project managers would have more flexibility when using the program. The second development concerns the better understanding of the results obtained and the explanation behind it. It would be interesting to understand in which percentage the discrepancy between solutions is due to a better program's solution, and in which depends on the fact that the models still need enhancements.

REFERENCES

Abara, J. (1989). Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces*, *19*(4), 20–28.

Aboudi, R., & Nemhauser, G. L. (1991). Some Facets for an Assignment Problem with Side Constraints. *Operations Research*, *39*(2), 244–250.

Bellenguez-Morineau, O. (2008). Methods to Solve Multi-Skill Project Scheduling Problem. *4OR*, *6*(1), 85–88.

Bellenguez-Morineau, O., & Néron, E. (2007). A branch-and-bound method for solving multi-skill project scheduling problem. *RAIRO - Operations Research*, *41*(2), 155–170.

Bellenguez, O., & Néron, E. (2005). Lower Bounds for the Multi-skill Project Scheduling Problem with Hierarchical Levels of Skills. In E. Burke & M. Trick (Eds.), *Practice and Theory of Automated Timetabling V* (pp. 229–243). Springer Berlin Heidelberg.

Caron, G., Hansen, P., & Jaumard, B. (2019). The Assignment Problem with Seniority and Job Priority Constraints. *Informs*, *47*(3), 449–453.

Cordeau, J. F., Laporte, G., Pasin, F., & Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, *13*(4), 393–409.

Klingman, D., & Ross, G. T. (1973). Finding Equivalent Network Formulations for Constrained Networks Problems. *Informs*, *23*(7).

Krishnamoorthy, M., Ernst, A. T., & Baatar, D. (2012). Algorithms for Large Scale Shift Minimisation Personnel Task Scheduling Problems. *European Journal of Operational Research*, *219*(1), 34–48.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics*, *52*(1), 7–21.

Kumar, A. (2006). A modified method for solving the unbalanced assignment problems. *Applied Mathematics and Computation*, *176*(1), 76–82.

Mazzola, J. B., & Neebe, A. W. (1986). Resource-Constrained Assignment Scheduling. *Operations Research*, *34*(4), 560–572.

Meneghetti, A., & De Zan, E. (2016). Technicians and Interventions Scheduling for the Maintenance Service of Container Ships. *Procedia CIRP*, *47*, 216–221. https://doi.org/10.1016/j.procir.2016.03.078

Pathan, S., & Shrivastava, V. (2021). *Reinforcement Learning for Assignment Problem with Time Constraints*.

Rabbani, Q., Khan, A., & Quddoos, A. (2019). Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation, 361*, 493–498.

Smet, P., Wauters, T., Mihaylov, M., & Vanden Berghe, G. (2014). The Shift Minimisation Personnel Task Scheduling Problem: A new hybrid approach and computational insights. *Omega*, *46*, 64–73.

Appendix I

ENGINEERS ALLOCATION – HANDBOOK

The purpose of this handbook is to give a step-by-step view to users of the "Engineers Allocation – ISEG & Locktel Project" program. This program arose from a agreement between ISEG and Locktel Ltd. under a student's – Raimundo Ramos – Masters Final Work.

After a careful reading of this handbook, every user will be able to fully benefit from the program advantages in the process of allocating engineers in Locktel's framework.

A. In BigChange

1. **Export Resources**: My Account → Administration → Resources → Add-edit → Export CSV with "ISEG-LOCKTEL project b1" blueprint.
2. **Export unscheduled tasks**: Schedule → Display → List → Choose filters if you like (unallocated and unactioned compulsory) → Export CSV without blueprint.

Save both documents in the same folder as "Engineers Allocation – ISEG & Locktel Project" and "OpenSolver" excels.

B. In Engineers Allocation - Nuno FINAL

1. **Enable Macros.**
2. **Import data**: "IMPORT" button → select wanted workers (Engineers Scotland) → select wanted clients (name of client than "ok" button or "cancel" button if nothing to add).
3. **Select Jobs:** "Select Jobs" button → "Select" button in every job desired → "back" button
4. **Compute export:** "Compute" button.
5. **Check and change if needed output** → then "Export" button.

C. In BigChange

1. **Import scheduled tasks:** Schedule → Display → List → Import without blueprint → Select excel file named "BigChange" in the same folder as "Engineers Allocation – ISEG & Locktel Project" and "OpenSolver" excels.

Every time this appears, click on "End" button, and contact Raimundo Ramos in order to understand or repair the problem if needed.

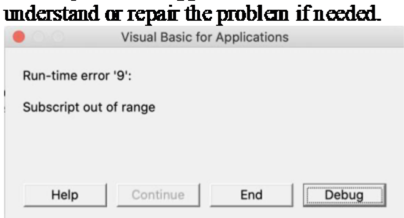

Figure 10 – Program Handbook

Appendix 2

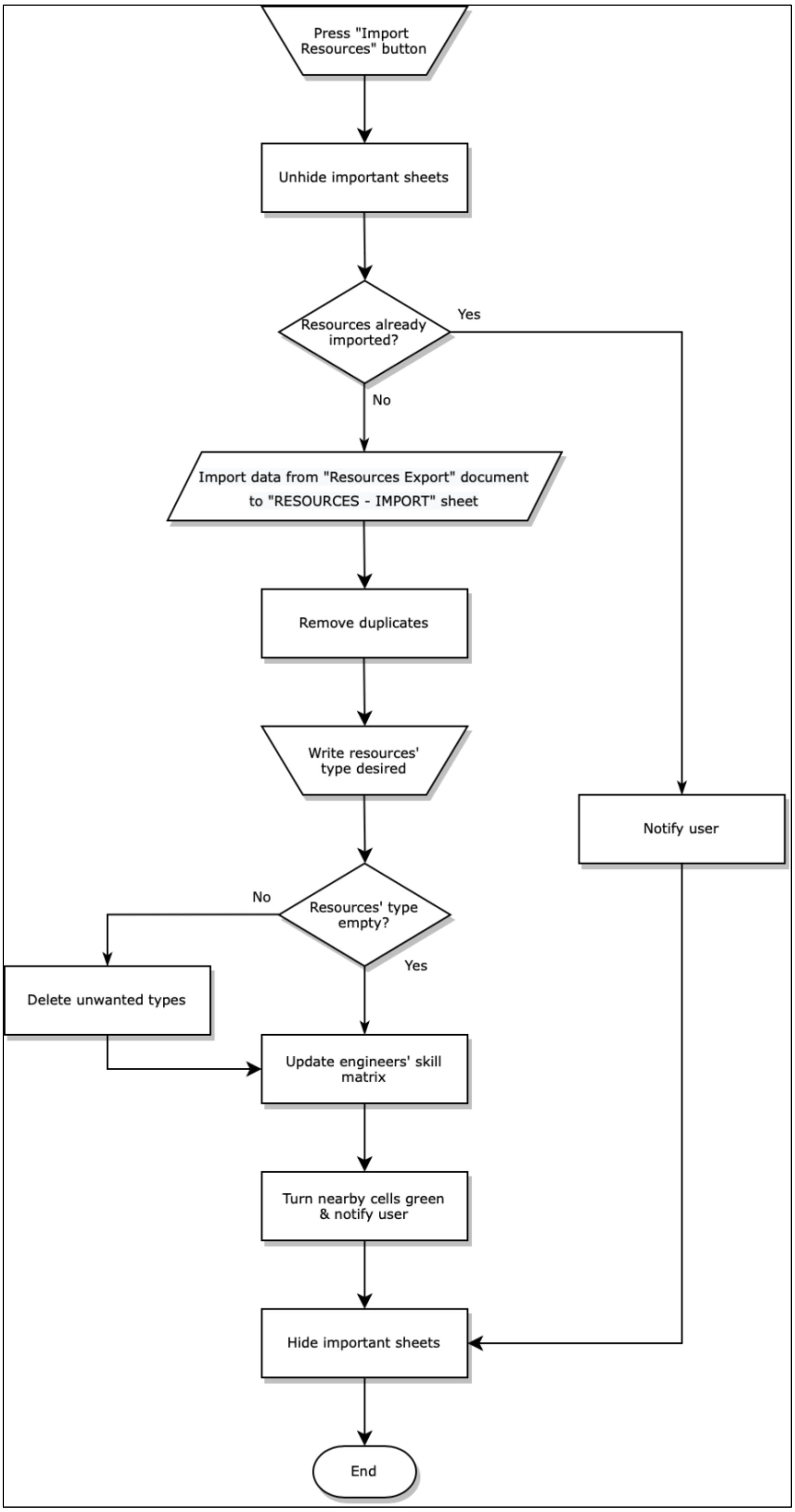

Figure 11 – Import Resources Macro Flowchart

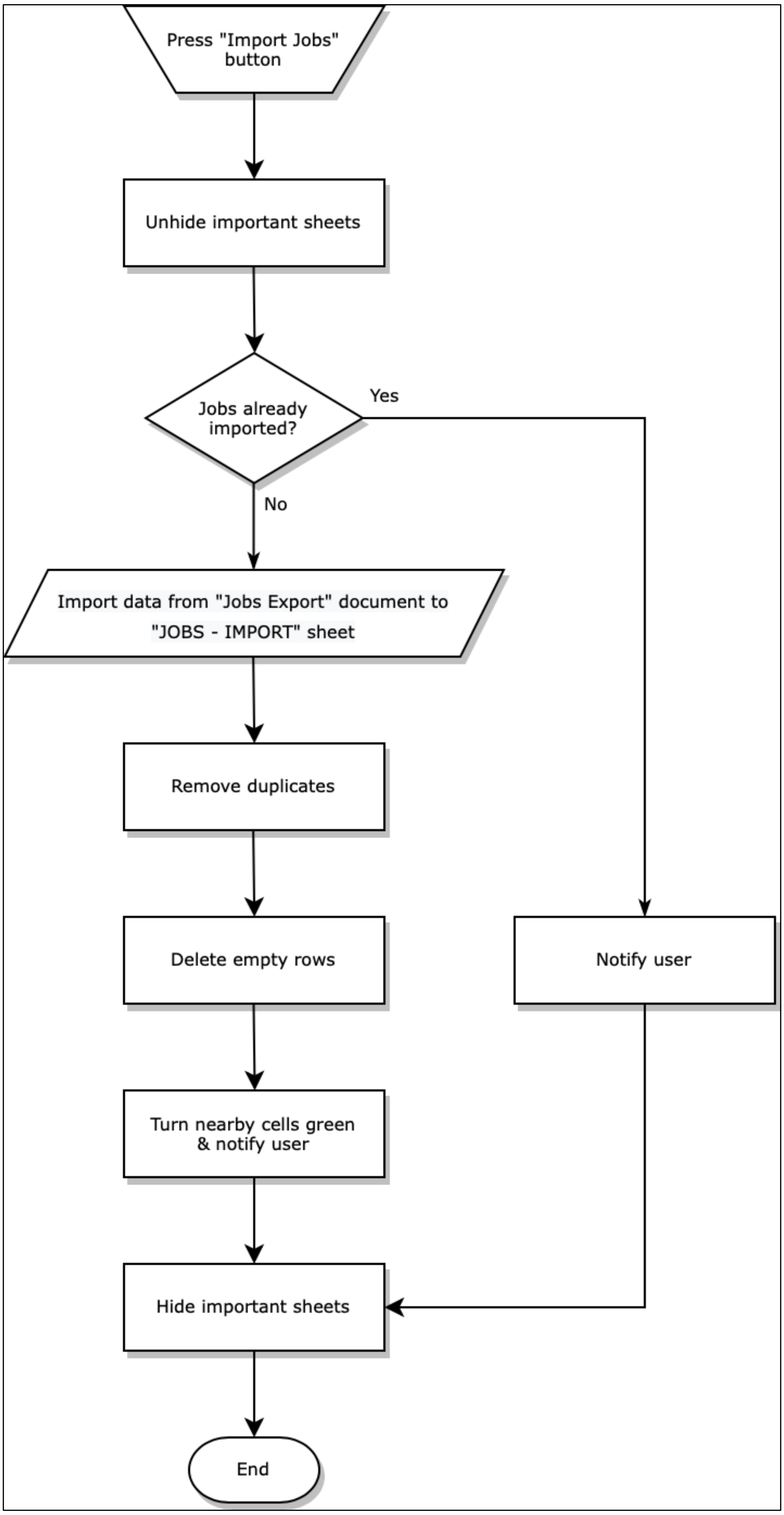

Figure 12 – Import Jobs Macro Flowchart

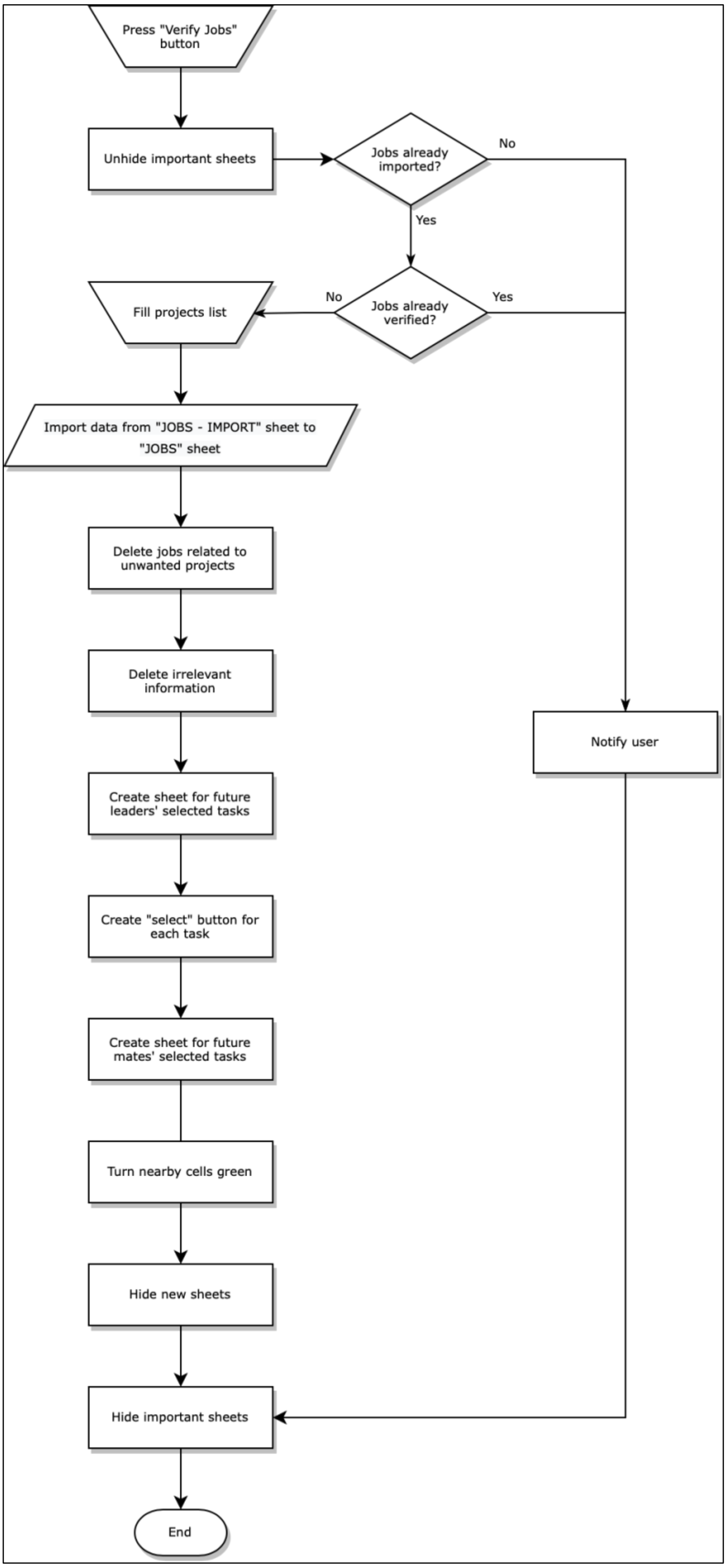

Figure 13 – Verify Jobs Macro Flowchart

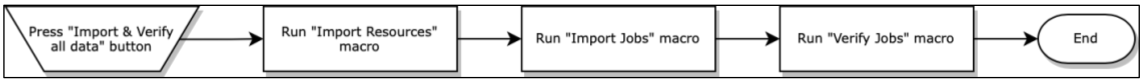

Figure 14 – Import & Verify all data Macro Flowchart

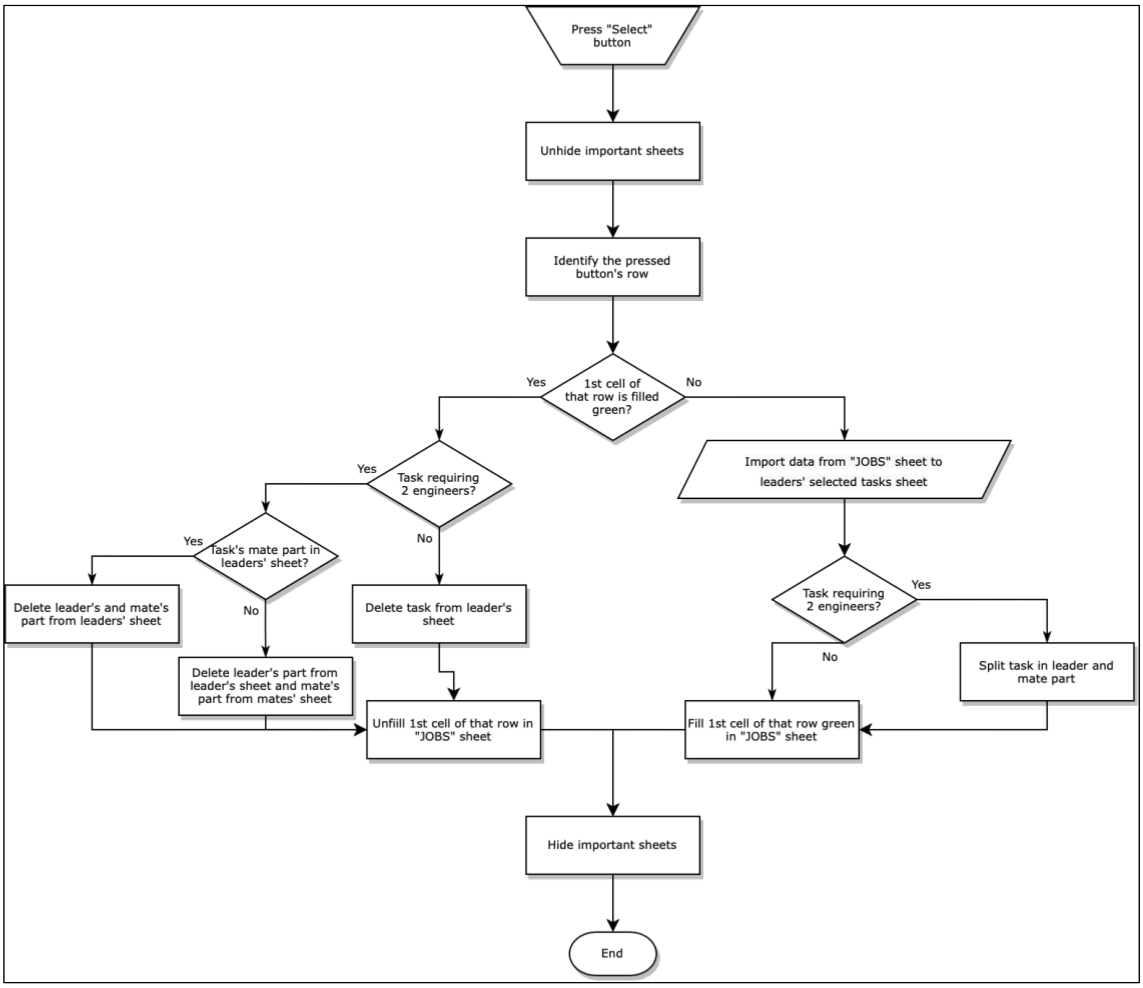

Figure 15 – Select Macro Flowchart

Figure 16 – Compute Module Macro Flowchart

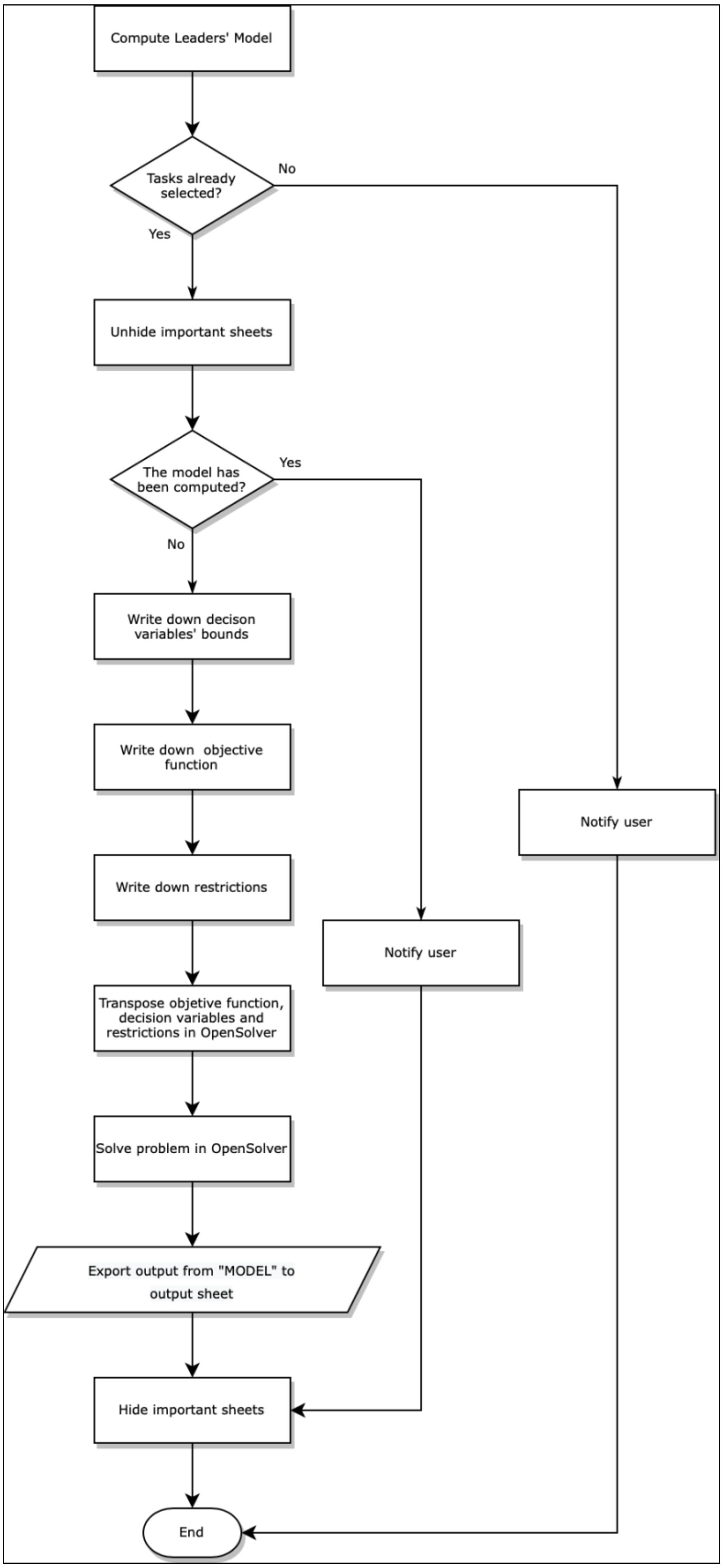

Figure 17 – Leaders' Model Macro Flowchart

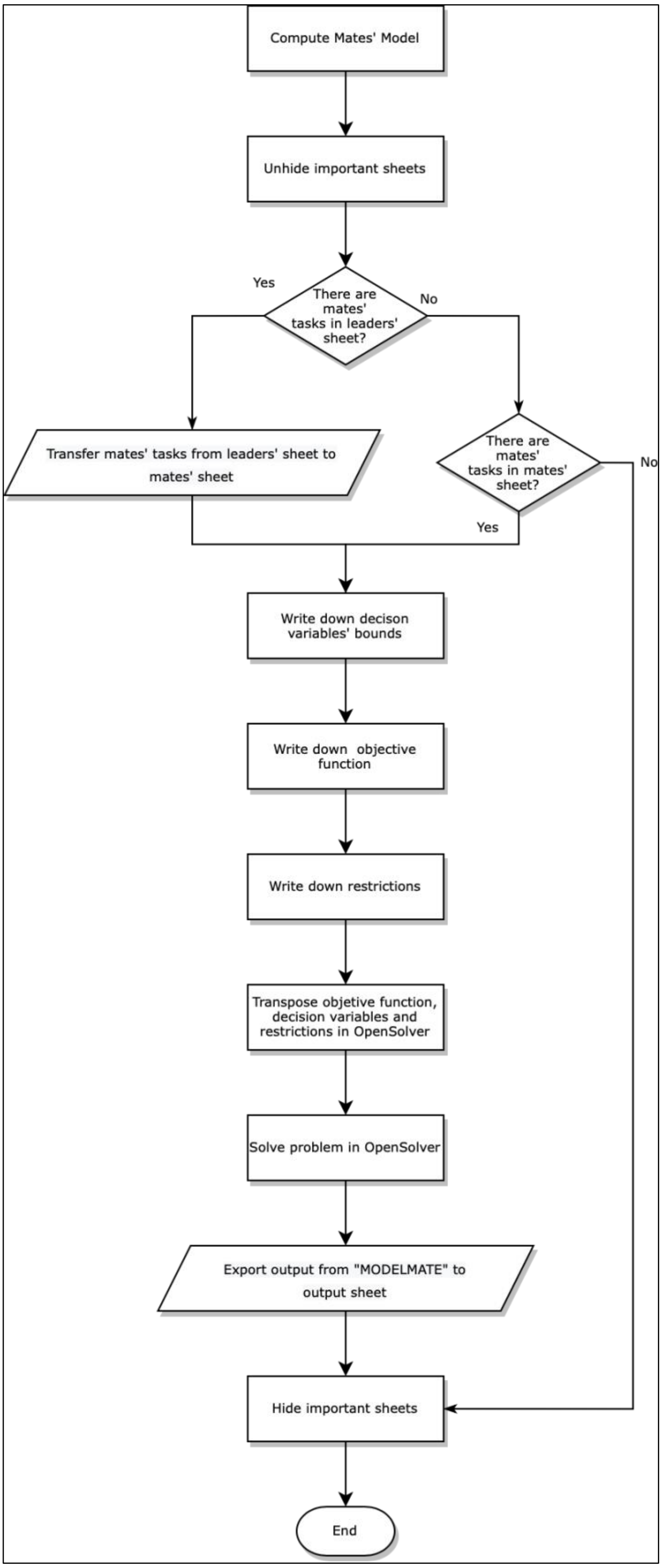

Figure 18 – Mates' Model Macro Flowchart

Appendix 3

<table>
<tr><th>Engineers</th><th>08:00 - 09:00</th><th>09:00 - 10:00</th><th>10:00 - 11:00</th><th>11:00 - 12:00</th><th>12:00 - 13:00</th><th>13:00 - 14:00</th><th>14:00 - 15:00</th><th>15:00 - 16:00</th><th>16:00 - 17:00</th></tr>
<tr><td>e1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e5</td><td>u7</td><td></td><td>u8</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e11</td><td colspan="9">u11
u12</td></tr>
<tr><td>e12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e13</td><td>u1</td><td></td><td></td><td></td><td></td><td>u2</td><td></td><td></td><td>u3</td></tr>
<tr><td>e14</td><td>u7</td><td></td><td>u8</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e17</td><td>u1</td><td></td><td></td><td></td><td></td><td>u2</td><td></td><td></td><td>u3</td></tr>
<tr><td>e18</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e19</td><td></td><td></td><td>u4</td><td>u5</td><td>u6</td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e22</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e23</td><td>u9</td><td></td><td></td><td></td><td></td><td></td><td>u10</td><td></td><td></td></tr>
<tr><td>e24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e25</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e26</td><td colspan="6">u13</td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e36</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e47</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e54</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figure 19 – 1st scenario, Solution A

<table>
<tr><th>Engineers</th><th>08:00 - 09:00</th><th>09:00 - 10:00</th><th>10:00 - 11:00</th><th>11:00 - 12:00</th><th>12:00 - 13:00</th><th>13:00 - 14:00</th><th>14:00 - 15:00</th><th>15:00 - 16:00</th><th>16:00 - 17:00</th></tr>
<tr><td>e1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e3</td><td>u8</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e6</td><td>u9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e11</td><td colspan="9">u11</td></tr>
<tr><td>e12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e13</td><td>u7</td><td>u3</td><td></td><td></td><td></td><td></td><td></td><td>u2</td><td>u1</td></tr>
<tr><td>e14</td><td>u8</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e17</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e18</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e19</td><td>u6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>u5</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e22</td><td>u4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e23</td><td colspan="6">u13</td><td></td><td></td><td></td></tr>
<tr><td>e24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e25</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e26</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e36</td><td colspan="9">u12</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e47</td><td>u7</td><td>u3</td><td></td><td></td><td></td><td></td><td></td><td>u2</td><td>u1</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>e54</td><td>u10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figure 20 – 1st scenario, Solution B