Rational Intelligent Project Scheduler for Software Project Management

A dissertation submitted for the Degree of Master of Information Technology

G.D.P.M. Handunge

University of Colombo School of Computing

2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: G.D.P.M Handunge

Registration Number: 2015/MIT/017

Index Number: 15550172

[illegible]

Signature:

Date: 22/03/2018

This is to certify that this thesis is based on the work of

Mr./Ms. G.D.P.M. Handunge

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Prof. N. D Kodikara

[illegible]

Signature:

Date: 22.03.2018

Abstract

Software systems have become a vital part in our current society and all of us are getting more dependent on services getting through many different software systems. Nevertheless, developing a good software is always not an easy task and making it a successful business is always a challenge. Among many problems in the software industry, one key problem is how to identify a good software project plan for a given software project? Many software development companies get project from different customers and they have to accurately predict the cost and timeline to make a good bid due to the higher competition in the market. Therefore, making a right software project plan is crucial for the software industry.

Software project scheduling is a complex task mainly due to the factors (e.g., time, resources, skills, holidays, etc.) affect on creating a plan. In computer science, this is an optimization problem and it is always a hard problem to find a correct schedule. Nevertheless, Genetic Algorithm (GA) has been identified as a best fit to handle this problem with a higher accuracy. GA being an evolutionary process, it provides an iterative process such that a population will lead to a convergence to achieve the required properties. Therefore, GA based Software Project Scheduling (SPS) is a key research area.

This research provides an approach for the research question: how to generate a project schedule for a complex software project as an automated task? This approach includes a series of steps that leads to solve this problem. It was identified the key factors that affect on SPS. First a software project schedule solution is encoded into a binary form. Having binary representation for the real problem a GA process is followed with rank section and elicit selection on a population. This process provides new individuals and each individual is quantitatively measured how quality it is. Having calculated the fitness values of each individual, higher valued candidates were selected for the new population and applied the GA operators (crossover and mutation) to derive new population. Those candidates' fitness also evaluated and this steps executed as an iterative process until a high quality solution is derived.

This approach was evaluated with Turing Test mode and identified the quality of the proposed methodology with GA, It is found that schedules derived through the developed prototype is comparable with what human experts created. Therefore, it is observed that the proposed approach is a good solution for software project scheduling, while some problems identified which needs to address in future work.

Acknowledgement

I would like to express my deepest gratitude to many people who helped to bring this project to fruition. First and foremost, I wish to thank my supervisor, Prof. N.D. Kodikara. I am so deeply grateful for the help, professionalism and valuable guidance throughout this project.

Finally, I must express my profound gratitude to all teaching staff, my family and friends for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

Thank you.

Table of Content

List of Figures

List of Tables

List of Abbreviations

SPS: Software project schedule

IT: Information Technology’

NP-hard: Nondeterministic Polynomial time hard

GA: Genetic Algorithm

ACO: Ant Colony Optimization

AHP: Analytic Hierarchy Process

SA: Simulating Annealing

CPM: Critical Path Method

MAOA: Multi-agent Optimization Algorithm

Chapter 1: Introduction

Software project management is both an art and a science for planning and driving software projects. Software project managers are responsible for executing accurate, and efficient project management, and it is a critical task with higher responsibilities. Success of a software project highly correlated with its project plan. If a project plan consists with issues, it is not possible to expect good results throughout the process. Process of creating a schedule for a software project by considering all the resources and tasks required is referred as software project scheduling. An accurate software project schedule (SPS) facilitates a logical schedule that provides sequence of activities that need to be performed within a given time frame while minimising the project cost by using the required resources in an efficient and effective manner [1–3]. Though many ingredients are required for a success of a software project, a poor SPS always leads to a project failure in many situations [4]. Therefore, generating a proper SPS is a task of an expert though it has many difficulties to achieve.

1.1 Background and Motivation

In the current socioeconomic context, human needs and lifestyles have become more complex and complicated. In contrast, people's aspirations to achieve a simple and happy life is getting stronger too. People expect much easier and productive systems that they can use to achieve their goals and needs effectively, productively, and economically. Among the limited options available to develop such systems, a promising approach is Information Technology (IT) based solutions [5]. Almost every industry uses software systems to do their work effectively and efficiently. Software development related businesses have become a new trend in the current market economy and billions of dollar investments are on it [6–7]. Nevertheless, it is difficult to effectively handle these requirements with traditional IT approaches efficiently [1–2], [8–9].

Delivering a qualitative, economical, and productive IT system is not a trivial task [1]. This is specifically due to higher demand, competition, limitations in technologies, and a lack of skilled people in the software industry [1–4], [6–8]. These problems and challenges are exponentially increasing and forming a bottleneck in the software industry. Therefore, we need different techniques and methodologies to cope with current demands in the software industry, to maintain good quality and lower cost [10]. How to address this problem as a whole is a broader research question, and under that umbrella, the question of how to generate a rational project schedule for a complex software project as an intelligent, automated task is a fundamental research question for all the people in IT industry. A correct project schedule always facilitates a good working plan with milestones to achieve ultimate goals [1]. Nevertheless, in today's context, making a rational project schedule in an intelligent manner can no longer be handled manually [11].

To come up with a proper schedule to develop a complex system, it requires to scrutinize the problem thoroughly and divide it into possible atomic tasks [2–3], [12–13]. Having such a collection of atomic tasks, it is necessary to identify dependencies among those tasks. Additionally, it is required to estimate the time needed to complete each task. In parallel to this, it is required to identify specific challenges associated with each task and the software development skills required to accomplish it [3], [12–13]. If the problem is not properly analysed, it may lead to a premature project plan. Therefore, the project will lag and resources will be wasted. Having all of these basic ingredients correctly, it should be possible to make an accurate project schedule/plan [11]. Nevertheless, it is a question that what is the most rational way to combine all of these tasks such that to achieve project goals. In addition, a good project plan should facilitate a good working schedule that contains information about major milestones that are bound with sub goals. Furthermore, a project plan should minimize the risk of a project failure [1–2].

The majority of problems given to software companies are related to complex systems or sub-problems of complex systems. Inherently complex systems include large numbers of diverse components that are engaged in unpredictable interactions. Additionally, its global properties emerge from the interaction of constituent components which are not present in any of those components. Furthermore, a complex system has a variety of ways of achieving the same or similar results, and therefore, there is no specific path to achieve the same quality results. With all these constraints, limitations, and errors with humans, it is not a naive task to make a rational project plan for a complex system by manually.

Rationality is the quality or state of being reasonable, based on facts or reasons. This allows to cope with higher computational time required to find the best solution. The trade-off between finding the best solution and an optimal solution in terms of computational time is huge, though the quality of the results is insignificant. Therefore, it is required to explore techniques to make a rational project plan for such a system. This particular problem can be condensed as combinatorial optimization problem and many optimisation methodologies can be included to achieve quality results [14]–[16].

Software systems provide solutions to complex systems; therefore, complex problems in software project management may also be addressed through new software systems.

Developing an effective and rational project scheduler for a complex system can also be considered as another software system. Project managers put lot of time and effort on planning a project schedule. This being a manual and complex task, it is hard to consider all the constraints and sufficient amount of options in parallel. Therefore, in most of the situations project schedules that are planned by manually are not the best. Therefore, there is a higher demand in the industry for intelligent tools that can facilitate a project plan when given required constraints. Nevertheless, such tools are very expensive. Therefore, the proposed system should be intelligent too. System should have the knowledge to proactively explore the search space sufficiently and to find a solution without the human intervention. Being this solution comparable with the features of a solution that human expert manually identified, this can be used to measure the level of intelligence of the system.

1.2 Aim and Objectives

The main aim of this project is to generate a rational project schedule as an intelligent system for software project management. Additionally, the main research question is how to generate a project schedule for a complex software project as an automated task. Having this aim with this research question, many objectives are identified to fulfil when completing this project. Those are namely as follows:

- To provide a literature review of various approaches on project scheduling:
  It is required to explore the current state of the research related with project scheduling. This provides to get a good understanding of what are the strengths and weaknesses of each methodology and to get a good intuition to develop a working methodology for project scheduling.
- Identifying the most important factors affect on software project scheduling:
  There may be many factors that may important in software project scheduling. It is important to identify which are the most important factors in SPS in general. As an example, in Sri Lanka, number of working hours of a software developer is not always 8Hrs per day, therefore, contribution from an employee may be more than 100%.
- Developing an approach for intelligent project scheduling:
  As the main technology, genetic algorithm (GA) will be used, and how GA can be used in a project scheduling is the main outcome of this objective.
- Implementing a working prototype:
  The proposed approach will be implemented as a prototype by using a suitable programming language (for e.g. in Java). This prototype should use set of constraints

specific to a software project as inputs together with other required input data (for e.g. some parameter values). Having this input data, it should generate a Gantt chat which is the standard format in software industry for a project schedule.

- Provide a method to input project data into the system and to presents the output data
  Through this it should be possible for a project manager to interact with the system easily.
- Evaluate the quality of the system
  System will be tested with actual data of a real software project. This provides the confidence how good the proposed system is and considered as a formal project evaluation.
- Interpret and analyse the achievements with the discussion for future work.

1.3 Scope of the work

The scope of this project mainly condensed to a working demonstrable prototype that has the potential to provide a rational intelligent project schedule when given necessary information. In practical context it is computationally a difficult task to find the best project schedule for a complex system and it may take a lengthy time to derive a solution. Therefore, in the scope of this project the main expectation is to obtain a rational project schedule (in this context rational means: a solution which is logically justifiable though it is not the best solution) than the best project schedule. Therefore, having this feature, the proposed system has the ability to stop the searching on a large search space once it is found an optimal solution which can be logically justifiable.

In addition to the rationality system is expected to be intelligent as well. In this context, what is intelligent means that is the ability to find a good solution as an autonomous process where there is no human intervention. Nevertheless, at the beginning a user should need to provide the necessary data and constraints of the software project as an input. Furthermore, finding the right parameter values will not be considered as the part of the intelligence of the system.

This project is considered to be as a research project than a software development IT project. The main reason why this is called as a research project is that there is no specific approach that always provides a rational project schedule intelligently. AI related techniques have been used for this purpose though it is difficult to identify a suitable approach considering all the specific features required (for e.g. task dependencies, resource availability, skill availability, work schedule of employees with flexibility to work overtime, time constraints, etc.). In this context, there is a room to explore a suitable methodology and putting all these constraints into a one specific approach to design and develop a system that provides logically acceptable, interesting results.

1.4 Assumptions

Following are identified as assumptions as this research:

- Given project constraints won't be change in the middle of the project, or if it is changed, the schedule should be regenerated.
- It is assumed that data which will be provided as inputs are correct and accurate.
- Standard computer hardware (processing power and RAM capacity) available are sufficient enough to execute GA based application.
- Intelligence can be measured through comparing the quality of an output generated by system and a human.
- It is not required to generate project schedules for multiple projects in a single execution.
- People will provide correct inputs for the questioner provided.

1.5 Thesis outline

The thesis is organized as follows. This first chapter provides an introduction to the research including a motivation, an aim, objectives, together with project scope. In chapter 2 a literature review is provided that includes state of the art information of software project scheduling. It followed by chapter 3 as a methodology for an innovative application. This mainly included genetic algorithm based methodology to generate a project schedule. Chapter 4 is provided implementation details about the prototype of proposed methodology. After detailed information of implantation details, chapter 5 is provided evaluation methodology of the system and results collected. Chapter 6 is included a conclusion and statement about future work. Finally, thesis is included with reference list and necessary additional information as appendices.

Chapter 2: Literature Review

2.1 Introduction

In the process of software development, a project schedule gets significant attention. The project schedule communicates what work needs to be performed, who will work on each task, and the timeframes in which that work needs to be performed. The project schedule should reflect all of the work associated with delivering the project on time. Without a full and complete schedule, the project manager will be unable to communicate the complete effort in terms of cost and resources necessary to deliver a project. Furthermore, software development is not analogical to a manufacturing process of traditional product development. In those processes, it is very clear what to do and how to implement. Also, including more workers and resources they can speed up the work in most cases. In contrast, software is often a process of discovery and increasing number of engineers will not solve problems if they encounter poor scheduling. Therefore, it is very important to create a rational software project schedule intelligently, and many researchers are working on this.

2.2 SPS with ant colony optimization

Xiao, Ao, and Tang [17] prepossessed an ant colony optimization (ACO) [18] based approach (which they called as ACS-SPSP algorithm) for SPS. They have highlighted usefulness of ACO on solving graph-based search problems [18–21], which is considered to be a one form of representation for SPS. In this regard, by splitting the tasks and distributing employee dedications to task nodes, they have formed a construction graph. Additionally, they have used the domain knowledge as heuristics to enhance the search ability of ants [19]. Six heuristics are used in ACS-SPSP algorithm, including 1) total dedications in tasks, 2) allocated dedications of employees to other tasks, 3) importance of the task in project, 4) both the total dedications in tasks and the allocated dedications of employees, 5) both the total dedications in tasks and the importance of the task in project, and 6) a constant. These heuristics are generic and useful to consider different aspects of the trade-offs in SPS by agents. They have compared ACS-SPSP with a genetic algorithm (GA) [22] implementation to solve the PSP. Furthermore, based on their results they have claimed that proposed algorithm is better when compared with GA based techniques [17]. Nevertheless, it is a question that the configuration parameters that they used in GA are the best to achieve the results [22], and also whether the given problems are complex enough such that available heuristic information are not directly leads to a solution with ACO [23]. Furthermore, they have highlighted the importance of extending their work to a multi project software scheduler.

2.3 Multi-project scheduling with priority rules and analytic hierarchy process

Singh [23] has presented a hybrid algorithm that integrates the project priority with project schedule development for multi-project scheduling. The main objective of this work is how to minimize the project make-span and the penalty cost when some projects carry higher priority. He has solved this problem by integrating the project priority with the activity priority, such that the compound priority has been considered in the decision making process. As the methodology, a project schedule is generated using a hybrid algorithm which is based on priority rules and analytic hierarchy process (AHP). The AHP is a structured technique for organizing and analysing complex decisions, based on mathematics and psychology [24]. Singh's methodology includes 7 steps [23]:

1. Analytic hierarchy process (AHP) initially divides a complex multi-criterion decision-making problem into a hierarchy of interrelated decision criteria, and decision alternatives.
2. A comparison among the alternatives and criteria is then made.
3. Compute the weightage decision matrix.
4. Repeat step 2 and 3 for each criteria and compute the weights of the projects with respect to all considered criteria.
5. Compute the priority index of each project.
6. The priority index of each projects computed in step 5 is used in combination with the activity priority index computed by the best reported dispatching rule to decide the priority of the project activities for resource allocations.
7. The performance of the project schedule is measured in terms of the make-span. Make-span of a project is the difference between the completion date of last activity and start date of the project.

2.4 Simulating Annealing based project scheduling

Bouleimen, and Lecocq [25] used Simulating Annealing (SA) [26] approach for generating the schedule of resource constrained project problems. Furthermore, [27] has also used SA based approach for generating the schedule of resource constrained multiple project problems similarly as Singh [23]. SA is often characterised by fast convergence and ease of implementation and therefore has found as useful to use in project scheduling. In this process as steps: problem encoding, neighbour generation, and applying a cooling scheme has been used. The computational results were compared with 20 existed priority rules in terms of project completion time [27]. They claimed that the simulated annealing approach yielded better results than priority rules.

2.5 Software project scheduling with Bayesian Networks

Khodakarami et al. [28] has claimed that basic inputs to a software project scheduling problem are not deterministic and those affected by many sources of uncertainty. Therefore, they have presented an approach by using Bayesian network modelling, that addresses both uncertainty and causality in project scheduling. Their proposed model uses Critical Path Method (CPM) to handle uncertainty. Though CPM is an interesting and simple model, its single point estimate assumption unable to deal with higher order complexities in real software projects. Therefore, they have done a translation from CPM to a Bayesian network to deal with complexities but the basics from CPM. Through this they have achieved followings (see [28]):

- *Capture different sources of uncertainty and use them to inform project scheduling.*
- *Express uncertainty about completion time for each activity and the whole project with full probability distributions*
- *Model the 'trade-off' between 'time' and 'resources' in project activities*
- *Use 'what-if?' analysis for finding the level of required resources given constraints like, for example, a specific completion time*
- *Learn from data so that predictions become more relevant and accurate*

Nevertheless, the proposed approach by Khodakarami et al. was only applied to a relatively simple example and they have proposed a future work that includes using the so-called Object Oriented BNs.

2.6 Multi-Agent Optimization Algorithm for SPS

Zheng and Wang [29] have used Multi-Agent Optimization Algorithm (MAOA) for resource-constrained project scheduling problem. In this algorithm, each agent represents a solution while all agents work in formed groups/teams. Each agent's evolution is based on 4 factors namely: social behaviour, autonomous behaviour, self-learning, and environment adjustment. In this factors social behaviour facilitates global and local explorations. Each team has a leader agent and the leader agent in every group is guided by the global best leader for global explorations. In the meanwhile, each agent of a goal guided by its own leader agent for local exploration. Each agent is fully autonomous and through this, each agent exploits its own neighbourhood. With the self-learning, the best agent that performs a quality searching allows to further exploits the promising region proactively. Meanwhile, some agents perform migration among groups to adjust the environment dynamically for information sharing. They have experimented this MAOA with other 14 algorithms (see [29] for references of those algorithms) and observed that MAOA is relatively better in medium and large scale SPS problems.

2.7 Generic algorithm based SPS

Chang et al. [30] have presented a scheduling model based on genetic algorithm to find optimal and near-optimal solutions. Furthermore, they have conducted a comparison between GA and hill-climbing algorithm [31] to measure the performance between two. The hill-climbing algorithm first starts with a population of initial solutions, and it is chosen the best one as the initial starting point. This foremost solution is mutated at a randomly chosen single pints on the search space and the fitness is evaluated. If the current position leads to a higher fitness, the new solution replaces the old one, and this procedure continues until the optimum is found [31]. According to their analysis, hill-climbing algorithms are much faster than GA to reach optima, and hill-climbing is much better than GA both in time and fitness evaluation. Nevertheless, when the search space includes many local optima it is a difficult task with hill-climbing algorithm [30]. This also implies the many techniques able to handle SPS when the complexity of the task is lesser but once the complexity is higher most of the techniques are having problems to find a solution in an acceptable time interval.

Alba and Chicano [11] has presented a system for SPS by using GA. They have considered many factors in a software project schedule including: available employees, skills of employees, salary of employee, dedication of each employee, task breakdowns, task dependency, cost of the project, and duration of the project. By considering all these factors, they have developed a system that generates project schedule when given project specific details. Based on the interesting results that they have obtained, they claim that GAs are quite flexible and accurate for SPS, and an important tool for automatic project management [11]. They have used 48 different project scenarios that created for testing purposes and performed 100 independent runs for each project. As limitations and future work, they have highlighted the problems with complexity of dealing with a large team, the overhead of assigning a large set of tasks to an employee. Furthermore, they expect to optimize the methodology to handle duration and cost of the projects more intelligently. Also, they have heighted the impotence of validating the approach with real world data in order to illustrate the usefulness of the approach in a real software project.

2.8 Summary

The SPS is a combinatorial optimization problem and an exhaustive search can take unacceptable duration to find a solution [3], [11]. Therefore, it is necessary to explore techniques that are capable of finding a quality solution with an acceptable time interval [2–3], [32]. Many researches have used different techniques in this regard as highlighted in this chapter. Though some approaches have provided interesting results, still SPS is not a problem that has definite solution. Some reasons for this is the nature of the project and its complexity. Technologies are always including both advantages and disadvantages; nevertheless, as per the section 2 there is a proper evidence that shows genetic algorithm based approaches are playing leading role in terms of addressing the SPS problem [11–12], [30], [33–34]. Furthermore, it has observed that many techniques able to handle SPS when the complexity of the task is lesser but once the complexity is higher most of the techniques are having problems to find a solution in an acceptable time interval.

Scheduling requires the integration of many different kinds of data and scheduling problems are dynamic and are based on incomplete data [2], [31]. In general, scheduling problems are NP-hard, meaning there are no known algorithms for finding optimal solutions in polynomial time [31]. As a result, most research has been focused to either simplifying the scheduling problem such that it can find a solution in a reasonable time period, or to use strong heuristic knowledge for finding good solutions. Scheduling problems include many types of constraints including: temporal constraints, precedence constraints, availability constraints, and combinations. Therefore, it is necessary to conduct research to isolate quality results quickly when the problem size grows or when additional constraints are required. Table 2-1 summaries the literature findings of this chapter.

Table 2-1: Literature Summery

Reference	Technology	Advantages	Limitations
[17]	Ant colony optimization	• Accepting spitted tasks and employees' dedications • Forming a construction graph • Use domain knowledge as heuristics • Six generic and useful heuristics are used in the algorithm	• Whether this approach can be used in complex scheduling problems • Needs to extend for multi project software scheduling
[23]	Priority rules and analytic hierarchy process	• Integrates project priority with the activity priority • Supports multi project scheduling • Comparison among alternatives • Supports priority index for each project	• Difficulties in Analytic Hierarchy Process

[25], [27]	Simulating annealing	• Fast convergence and ease of implementation • Provides better results than priority rules	• Cost function is expensive to compute • Cooling must be very slow, therefore may take longer time to converge
[28]	Bayesian networks	• Includes uncertainty of sources • Uses causality in project scheduling • Considers the trade-off between time and resources	• Only applied to a relatively simple example • Expected to include object oriented Bayesian networks
[29]	Multi-agent technology	• Supports global and local explorations • Allows to have large number of agents to explore the search space thoroughly • Agents are autonomous and have the self-learning ability	• Needs protocols for agent communication • Needs more computing power if the total number of agents are higher
[11], [12], [30]	Generic algorithm	• Provides optimal and near-optimal solutions • Encoding is easy and can easily integrated features • Constraints can be integrated with fitness function • Less model parameters • Approach is nature inspired and easy to implement • GA is quite flexible	• Slow in finding the best solution and fitness evaluation • When the task becomes more complex and having many features, it may need longer duration to evolve to a good solution

Chapter 3: Problem Analysis and Methodology

3.1 Introduction

The main research question of this work is how to make a rational project schedule in an intelligent manner for the IT industry. As highlighted in the section 1.1 this is a very important requirement in the IT industry and many project managers are expecting suitable tools for this that make their decision making stronger, productive, and cost effective. Many researches have used interesting approaches and in section 2 some of those have highlighted. Each technique seems to have its own advantages and disadvantages, though GA based approaches have strong evidence to provide quality results and even potential to handle large projects. This section provides problem analysis and detailed methodology of genetic algorithm based SPS.

3.2 Problem Analysis

The main research question of this work is how to generate a rational project schedule as an intelligent system for software project management. In general, scheduling problems are NP-hard, meaning there are no known algorithms for finding optimal solutions in polynomial time. Therefore, SPS also suffers with the same set of challenges in NP-hard problems. Furthermore, project scheduling problems include many types of constraints including: temporal constraints, precedence constraints, availability constraints, and cost constraints. In addition to these constraints, it is required to include objectives also into software project scheduling. Having all these requirement, it is practically hard to find the best schedule for a software project. Therefore, the most rational decision is to find a high quality solution (which is referred as a rational solution) that can be accepted by a domain expert. Furthermore, this schedule should be automatically generated and intelligently.

The current selected problem includes many factors that needs to be considered. The major factors identified for this can be listed down as follows:

- Employees: Individuals who are working in a software project (software architects, tech leads, software engineers, quality assurance engineers, etc.)
    - ith employee denoted by: e_i, where i $\rightarrow$ 1 to E (E is the total number of employees of a given project)
- Skills: Skills of all the employees.
    - Skills of all employees denoted by set SK.For e.g.

$SK = \{java, oracle, regression, modling, \dots\}$

- Also a skill set of kth employee can be denoted as: e_i^{skills}. For e.g.

  $e_i^{SK} = \{java, oracle\}$

- Salary: Each employee has a salary.
  - ith employee's salary denoted as: e_i^{salary}
- Dedication: Each employee has a maximum degree of dedication to the project.
  - Dedication of ith employee is denoted by: e_i^{maxded}
- Tasks: A project includes n number of tasks to be achieved.
  - Tasks denoted by a set: $T = \{t_1, t_2, t_3, \dots\}$
- Task skills: To accomplished a task it has a set of required skills associated.
  - Denoted by t_i^{skills}. For e.g. $t_i^{skills} = \{regression, modling\}$.
- Effort for a task: Effort is the time required to spend by one person on a task and this is expressed in person-months (PM).
  - ith task effort is denoted by t_i^{effort}.
- Solution

  Having information that explain the nature of the software project it is essential to represent this schedule in a computational form. With all the above information a solution (which is the schedule) can be encoded as a matrix $Solution = (X_{ij})$ where the matrix size is: $Employees\ (E) \times Tasks\ (T)$, $X_{ij} \geq 0$, and the element X_{ij} is the degree of dedication of employee e_i to task t_j.

  - $Solution = \begin{pmatrix} 1.0 & 0.8 & 1.0 \\ 0.5 & 0.7 & 0.9 \\ 1.0 & 1.0 & 1.0 \\ 0.2 & 0.0 & 0.2 \\ 0.9 & 0.7 & 1.0 \end{pmatrix}$

In addition to these factors, these being a constraint satisfaction problem there are important constraints also identified to use in this approach. For a specific given software project under a particular context it is always difficult to tell what is the best schedule. Therefore, in this work the focus is given to a rational solution and having more than one such a solution these constraints provide the intuition to select which is as the best. It is possible to observe four form of constraints related with this problem [3], namely:

1. Temporal constraints
2. Precedence constraints

3. Availability constraints
4. Cost constraints

Temporal constraints are driven by time related effects. For example ith employee (e_i) works only on certain days only in the week. It is important to consider these special features in a large case software projects as most of the experts are not available in full time. Furthermore, this can include the time required to complete a software project. Having multiple rational schedules, it is possible to restrict the number of solutions by introducing time constraints. Some projects may require to finish on or before n number of months' time. Therefore, schedules that may take more than this limit can be considered as premature solutions.

Precedence constraints include the order of tasks. A software project can be considered as solving n number of atomic tasks; nevertheless, it is not possible to solve all of these tasks at the same time. In a software project, tasks include precedence and some tasks cannot perform in parallel. For example, system testing should start only if the unit tests are successfully completed. For this requirement task dependency graph can be used. A precedence graph provides the details about which tasks must be completed before a new task is begun. Precedence graph is an acyclic directed graph denoted by $G(V, A)$, with a vertex set $V = \{t_1, t_2, t_3, \ldots, t_T\}$ and an arc set A, where $(t_1, t_2, t_3) \in A$; which implies task t_1 should completed before tasks t_2 and t_3 starts (see Chapter 5 for an example).

Availability constraints are useful when employees have different dedications. Therefore, it is not possible to assume that all the employees will available anytime (this is different from temporal constraints and in their time is very specific and in here time is flexible but as a whole there are restrictions on availability, for example may available only for 2 months). As each employee is having a maximum dedication (e_i^{maxded}) it is always necessary to hold the constraint $e_i^{WL} \leq e_i^{maxded}$ (where e_i^{WL} is the *ith* employee's work load).

Cost is also very important for the competitive nature of the software industry. Therefore, always it is required to minimise the cost that emerges through selecting a specific schedule. Therefore, in this research cost factor also consider as a main contract. Low cost always encourages.

With each of these attribute definitions and constraints, it is possible to encode a software project in computational form. This knowledge representation is used as an input to the system. Therefore, it is necessary to provide all employees, skills of each employee, salary of each employee, dedication of each employee, software project task breakdown, skills required in each task, effort for each task and precedence graph for the project. Having these ingredients, the aim is to identify a process that is able to generate a rational project schedule as an intelligent system for software project management. Also, having many constraints, it leads to more computational time. In general, for a scheduling task, it may include many local optima (some of these considered as rational solutions); but, having many constraints, this limits to a specific few feasible solutions. This strengthens the quality of the solution, but increase the computational time to identify a solution. Therefore, having many constraints, it reduces the flexibility and therefore, it is good to balance this and select a quality solution which is both flexible and take relatively acceptable time to find the solution. This make this project a research challenge.

3.3 Research Methodology

Scheduling problems are considered to be optimisation problems and due to the nature of SPSP this has higher complexity and computationally expensiveness. Among many possible technologies explored that can be used for this problem (as in Section 2), genetic algorithm is identified as the rational choice (see Section 2.8). Genetic algorithms (GAs) were invented by John Holland in the 1960s [22]. GA is an evolutionary process that moves one population of chromosomes to a new population together with applying natural selection with genetic inspired operations namely crossover, mutation, and cloning. A chromosome is an encoded form of a solution that in a population.

A chromosome is an array of genes where each gene can be presented as a feature/property/attribute of a given solution. The selection operation provides which chromosomes to be selected from the population to apply for genetic operations. Depends on the selection method there will be different chromosomes that leads to fulfil different evolution paths to achieve a solution. Cloning, crossover, and mutations are the genetic operations used in GA. Cloning provides to duplicate the same chromosome in the next population. Crossover operation combines two different chromosomes and provides a new two chromosomes where the features of parent two chromosomes have combined in different form such that new solutions with different qualities. Mutation is a powerful operation that provides to invert the specific gene value of a chromosome. This leads to specifically improve (or lower) the quality of a single chromosome. Having new set of chromosomes through genetic operations from a new population, where it is required to evaluate the fitness of each new chromosome to decide the new population. Only the chromosomes with highest fitness values should go to the new population as per the evolutionary process: survives for the fittest [22]. This process iteratively evolves until finding a solution (i.e. a chromosome) which satisfy set of constraints that should hold in a goal searching.

GA are mainly used not to find the best solution but a solution which is strong enough but relatively quickly derivable [22], [33].In the SPS problem also the goal is not to find the best schedule but to find a rational solution (see Section 2), therefore GA satisfies this requirement and also provides more than one solutions that may be useful in some context of the project needs.GA being inspired by evolutionary process in the mother nature, depends on the complexity of the problem (which is proportional to the search space of the problem) the time this takes to find a solution may vary. Therefore, it is very important to execute GA process many times with different initial values and a random initial population. Figure 3-1 presents the overall process of GA and subsequent sub-sections of this chapter provides detailed information about each steps in terms of the problem elaborated in Section 3.2.

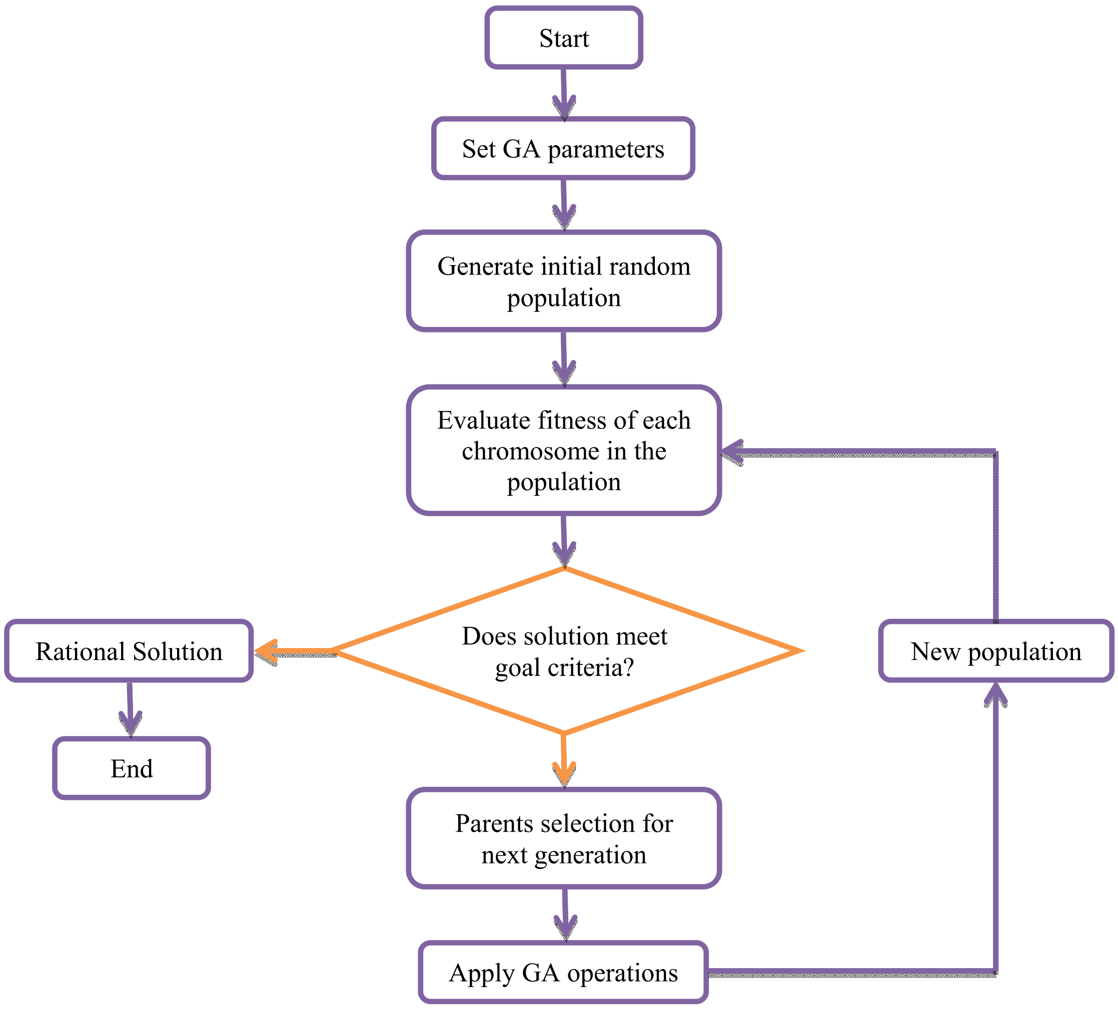

Start
Set GA parameters
Generate initial random population
Evaluate fitness of each chromosome in the population
Does solution meet goal criteria?
Rational Solution
End
New population
Parents selection for next generation
Apply GA operations

Figure 3-1: Genetic algorithm procedure for SPS

3.3.1 Set genetic algorithm parameters

GA being an evolutionary process its evolution depends on some parameter values used in the process. Mainly there are 3 parameters important in GA namely:

1. Crossover probability
2. Mutation probability
3. Population size

Crossover and mutation are genetic operators which are sensitive to the changes in population. Therefore, when applying these operators, it is necessary to do that systematically such that the population will quickly converge to a solution. For this in different frequencies (i.e., with different probabilities) crossover and mutation operations should perform in the GA process. In GA, crossover is performed with the hope of new chromosomes will have good features from the parent chromosomes such that new chromosomes will be better in fitness than its parents. Nevertheless, in an evolutionary process it is good to keep some relatively week features also in some chromosomes for later optimisations in the GA process. Crossover probability provides how often two parents should combine. Crossover is the main operation that provides to combine different features of two parents to form a new solution which may not in the previous population. Probability of crossover can vary from 0 to 1.0 where if it is 0 then the new generation will be the exact copy of the previous generation (assume that there is no mutation applied) as the only operator that can be used is cloning. Nevertheless, it is not possible to use crossover probability as 1.0 as it leads always to combined different parents and to form new offspring that will not leads to a convergence.

Similarly, mutation probability also can vary from 0 to 1.0, where if it is 0 there is no mutation and if it is 1.0 it is only the mutation. Furthermore, mutation is applied to either one or two genes of selected chromosome and this make sure that a solution will be changed slightly only. Mutation is mainly used in GA to getting away in tapping with local minima but should not occur very often as then it leads to a random search and never converge.

Population size is also another important parameter that will effect on the GA performance. It is necessary to have a good population size which is neither too large nor too small, nevertheless what should be the exact required population size is not something trivial. If the population size is too large, the GA process will be very slower due to the availability of many parents to apply GA operators. From research it has been shown that, if the population size is exceeding a threshold, there is no improvement to the convergence time [22] Similarly, if the population size is smaller, there are only few samples to select as parents and their distribution over the solution space is very limited. This will always lead to a searching on a small part of the search space of the problem.

De Jong [35] provided rules of thumbs for GA parameters' value selection. His research emphasised that the best single point crossover rate as ~0.6 per pair of parents, the best mutation rate as 0.001 per bit, and the population size as 50–100 individuals. As per Grefenstette's research [36], he proposed that population size to 30, the crossover rate to 0.95, the mutation rate to 0.01. When considering the difference of results of De Jong and Grefenstette, it is clear that for a smaller population and higher crossover and mutation rates De Jong's able to obtained interesting results in his work. After sometime Schaffer and colleagues [37] noted that population size as 20–30, crossover rate as0.75–0.95, and mutation rate as 0.005–0.01. These results clearly show that if the population size is small it is necessary to use relatively large probabilities for crossover and mutation. Nevertheless, when it comes to the most of latest research than sticking to specific values for these parameters researchers have used adaptive real-time methods that changes parameter values from beginning to end of the GA process (see [38]). In these techniques at the end of the process higher probability values will be used for crossover and mutation probabilities and over the time when the system is converging to a solution gradually these values will be very smaller.

3.3.2 Forming an initial population

In this approach it is essential to have an initial population which includes probable solutions. In GA a solution is always a chromosome and each solution should encode into a chromosome. As stated in Section 3.2 a solution (which is the schedule) of a give software project can be easily encoded as a matrix where rows and columns for employees e and tasks $t$and in each point the value is degree of dedication of employee e_i to task t_j. Therefore, solution encoding is very straightforward and a random sample population generator can be used for this. Random sample generator should accept the features mentioned in Section 3.2 and needs to generate random schedules (those may not be correct and even fully completed) properly selecting dedication value for a particular employee on a particular task. As an example following matrix can be used as a solution for a specific software project.

$$Solution = \begin{pmatrix} 1.0 & 0.8 & 1.0 & 1.0 & 1.0 \\ 0.5 & 0.7 & 0.9 & 0.9 & 0.9 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0.2 & 0.0 & 0.2 & 0.0 & 0.4 \\ 0.9 & 0.7 & 1.0 & 0.7 & 0.5 \end{pmatrix}$$

In this example each raw represent a specific employee and each column specifies a task where as a point on the matrix is the dedication of an employee on that task as required. Similar to this, multiple solutions can be represented to form a sufficiently large population.

Having this method feasible to encode a solution into a chromosome, having real numbers leads to difficulties with some GA operations (e.g. mutation where it is required to invert a bit value). Therefore, it is much easier to use the binary encoding in this regard. Due to this need it is required to transform these real values to its binary form. With the complexity of using real numbers to binary representation, and need of larger bit length to store a single real number, a simple approach can be used that preserves the accuracy too. Therefore, it is need to find max allowed working time of an employee. As per the rules this should be 8 hours but in the practical context in Sri Lanka this can be assumed as 12 hours for maximum. Therefore, finding this range it can be divided into 16 units such that each unit can be represented in 4 bits. Table 3-1 provides how dedication value should be represented in the binary form.

Table 3-1: Dedication degree and binary encoding mapping table

Real value (0-12h)	0	0.08	0.16	0.24	0.32	0.4	0.48	0.56
In Binary form	0000	0001	0010	0011	0100	0101	0110	0111
Real value(0-12h)	0.64	0.72	0.8	0.88	0.96	1.04	1.12	1.2
In Binary form	1000	1001	1010	1011	1100	1101	1110	1111

By using this table, it is easily possible to decide which binary value used for each dedication value. For example, assume that a given person is working 4Hrs per day (that means dedication is 0.4 as per the scale used) then its mapping binary value should be 0101.Similarly, each value can be transformed. For example:

$$Solution = \begin{pmatrix} 0.8 & 0.4 & 0.8 & 0.8 & 0.4 \\ 0.48 & 0.48 & 0.16 & 0.16 & 0.16 \\ 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \\ 0.16 & 0.0 & 0.24 & 0.0 & 0.4 \\ 1.04 & 1.04 & 0.4 & 1.12 & 1.12 \end{pmatrix} = \begin{pmatrix} 1010 & 0101 & 1010 & 1010 & 0101 \\ 0110 & 0110 & 0010 & 0010 & 0010 \\ 1010 & 1010 & 1010 & 1010 & 1010 \\ 0010 & 0000 & 0011 & 0000 & 0101 \\ 1101 & 1101 & 0101 & 1110 & 1110 \end{pmatrix}$$

3.3.3 Fitness function

Each solution in a population holds a specific fitness value. This fitness value is used to quantify the quality of each chromosome. Solutions with higher fitness values hold good quality solutions whereas the opposite holds the inverse. Therefore, it is important to identify a good fitness function that is capable of quantifying how far the solution is away from necessary constraints. For this following equation can be used as in [11]:

$$f(x) = \begin{cases} 1/q & if\ the\ solution\ is\ feasible \\ 1/(q+p) & otherwise \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \ (equation\ 1)$$

$$where: q = \omega_{cost}Proj_{cost} + \omega_{dur}Proj_{dur}\ and$$

$$p = \omega_{penalty} + \omega_{twne}Pro_{twne} + \omega_{reqsk}Pro_{reqsk} + \omega_{ow}Pro_{ow}$$

This equation represents the cost of a solution (i.e., q) and the penalty for unfeasible solution (i.e., p). To find a right solution with the constraints identified it is important to minimize both of these. Therefore, in this regard the goal is to find a solution that minimizes the p and q adequately. These equations consist with many arbitrary weight values and variables as follows:

- ω_{cost}: arbitrary constant for project cost
- $Proj_{cost}$: total cost of a solution
- ω_{dur}: arbitrary constant for project duration
- $Proj_{dur}$: total project duration of a solution
- $\omega_{penalty}$: arbitrary constant for project penalty
- ω_{twne}: arbitrary constant for the number of tasks with no employee appointed
- Pro_{twne}: total number of tasks with no employee appointed
- ω_{reqsk}: arbitrary constant for the number of skills missing in order to perform all project tasks
- Pro_{reqsk}: the number of skills missing in order to perform all project tasks
- ω_{ow}: arbitrary constant for overwork of the project
- Pro_{ow}: the total overwork of the project

To use this equation, it is important to assign arbitrary values and the same values suggested in [11] can be used as stated bellow.

- $\omega_{cost} = 10^{-6}$
- $\omega_{dur} = 0.1$
- $\omega_{penalty} = 100$
- $\omega_{twne} = 10$
- $\omega_{reqsk} = 10$
- $\omega_{ow} = 0.1$

Depends on the requirements whether the cost of the project or the duration of the project is the primary concern it is easily possible to achieve this through weigh values: ω_{cost}, and ω_{dur} respectively. To calculate the total $Proj_{dur}$ it is need to find the duration of each individual task (t_j^{dur}) as shown in equation 2. Having these for each task, further it is required to calculate starting and finishing times for each task separately. Together with this information $Proj_{dur}$ is maximum finishing time considering the task precedence graph data.

$$t_j^{dur} = \frac{t_j^{effort}}{\sum_{i=1}^{E} x_{ij}} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (equation\ 2)$$

Total project cost also mainly can be assumed as the salaries of employees (there are other cost also but those kept away from the scope of this work). Therefore, project cost can be calculated from the salaries to each employee assign to the project with their dedication to the work as in equation 3.

$$p_{cost} = \sum_{i=1}^{E} \sum_{j=1}^{T} e_i^{salary} x_{ij}\, t_j^{dur} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (equation\ 3)$$

Overwork required for a found solution is the last thing that needs to calculated. This is also can be calculated with a simple calculation. First it is required to find total dedication of each employee on the project (e_i^{work}). Having this for each employee if that is exceeding the maximum dedication of that employee (e_i^{maxded}), the difference is the overwork of that employee. This can be calculated as in the equation 4 and 5. Accumulating overworks of all the employees provides the total overwork of the project as in equation 6.

$$e_i^{over} = \int_{t=0}^{t=p_{dur}} ramp\left(e_i^{work}(t) - e_i^{maxded}\right)dt \cdots\cdots\cdots\cdots\cdots\cdots (equation\ 4)$$

$$ramp(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x\ \leq 0 \end{cases} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (equation\ 5)$$

$$p_{ow} = \sum_{i=1}^{E} e_i^{over} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (equation\ 6)$$

3.3.4 Termination criteria

Having calculated fitness of each chromosome in a population, it is important to decide whether there is a solution that satisfies the requirement. This requirement can be used as the termination criteria. This can be the maximum cost acceptable for the project and what equation 1 provides should less than this. Furthermore, this can be guarded with maximum allowed iterations for GA process and within that if the process unable to find that the best chromosome derived so far can be considered as the solution. Nevertheless, the last termination criteria are not to find a suitable solution but to deal with unacceptable time taken to find a solution. If GA takes unacceptable amount of time, the most logical step should be the change parameter values and execute the process again.

3.3.5 Parent selection methods

In GA process, depends on which parents are selected to apply for genetic operations, effects on final results and time needed to find a solution. Therefore, selection methods are very important in the GA process [22]. Appropriate selections contribute to have offspring that have even higher fitness than with individual parents. There are well known selection techniques available in GA namely: Fitness−Proportionate Selection, Elitism, Rank Selection, and Tournament Selection [22]. Each selection technique consists with different benefits.

Fitness proportionate selection is a most popular technique in GA and goes back to the Holland's original GA selection methods [22], [38]. In this method it is required to calculate fitness value of each chromosome (f_i) and the average fitness value ($\acute{f}$) of the population. Then the probability of i^{th} individual being selected (p_i) will be given by equation 7 where n is the total number of chromosomes in the population.

$$p_i = {f_i}/{n\acute{f}} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (equation\ 7)$$

Fitness proportionate selection provides reasonable distribution to select parents such that there is a high probability to select higher fitness candidates whereas less probability to select lower fitness candidates. This approach is mostly implemented by using the roulette-wheel basics. A roulette-wheel is a wheel that constructed dividing the 360° space into segments where the area is proportional to the fitness of each chromosome. Therefore, chromosomes with higher fitness values get larger area where as chromosomes with smaller fitness values get smaller area as shown in Figure 3-2.

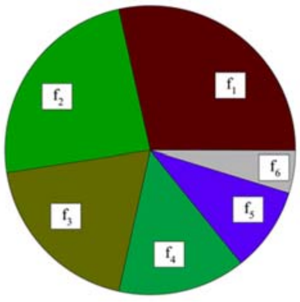


Figure 3-2: Roulette-wheel designed based on chromosomes' fitness values

When the wheel is spun, the probability of the roulette landing on f_i is p_i . Nevertheless, fitness proportionate selection suffers from premature convergence, stagnation [22], [38].

Rank selection also useful method for candidate selection. In this process all chromosomes are ordered in ascending according to their fitness values. Having this a probability distribution will be used mostly simple liner and exponential equations (see [22]). Having chromosomes bonded to a probability distribution the selection is also performed based on that probability distribution. The main problem with this method is this is not biologically justifiable.

Tournament selection provides good performance as it uses two steps to eliminate problems with previous methods. In this method chromosomes are divided into n number of groups (n>2) first. Having set of groups, the first is to select a group randomly. Having selected a group, the candidate with the highest fitness in that group will be selected as a parent. This process continues, and it is possible to form new groups also in the middle of the process [22]. Tournament selection inherent the advantages of rank selection but it does not require global reordering and it is more naturally-inspired.

Elicit selection is very useful selection technique not to use as the main technique but use in combined with all the previously mentioned selection techniques. Elicit selection ensures that at least one copy of the best individual in a population is always passed onto the next generation. This is very useful feature and always the best we found preserves.

In the SPS problems all these techniques can be used and tournament selection may provide interesting results depends on the other parameters.

3.3.6 Applying GA operations

Having selected parents, the next is to apply genetic operations. Mainly two operations are used crossover and mutation as defined in section 3.3.1. In this regard it is required to use probabilities mentioned in section 3.3.1 for crossover and mutation. Crossover needs two parents whereas mutation always with a single parent. In crossover it is possible to combine different features of each individual to form new individuals with single point crossover (Figure 3-3) and two-point crossover (Figure 3-4). Example of mutation for binary encoded chromosome has presented in Figure 3-5.

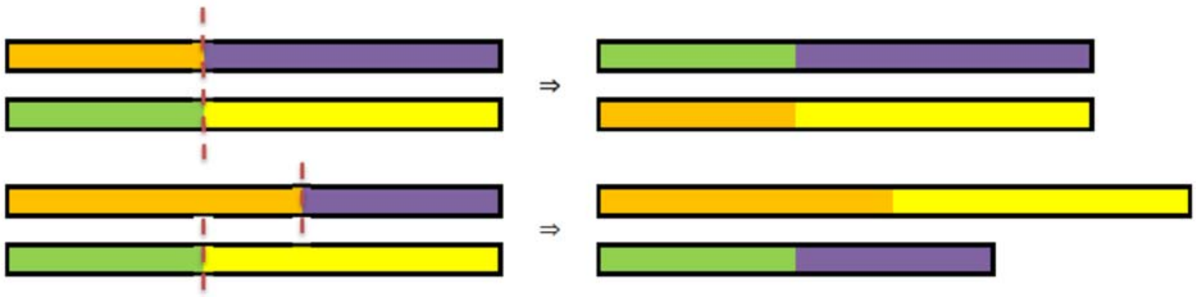


Figure 3-3: Single point crossover

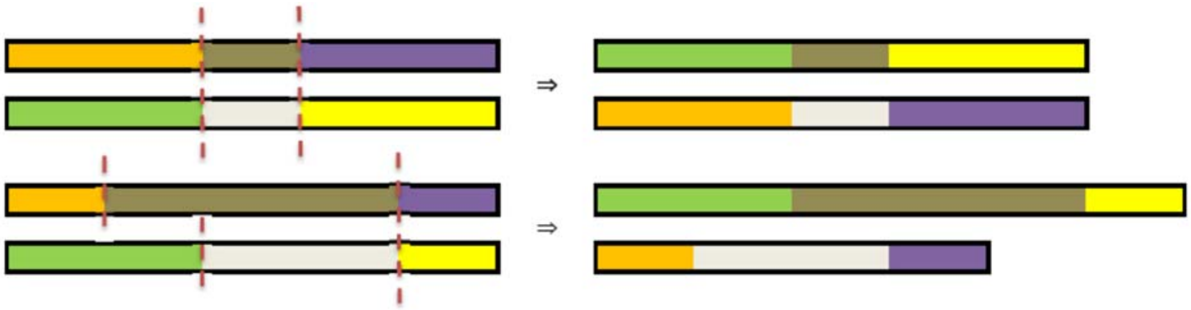


Figure 3-4: Two point crossover

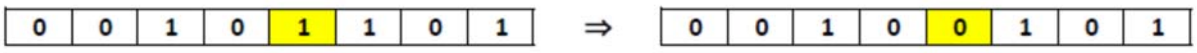


Figure 3-5: Mutation

The driving force of GA is genetic operators and depends on how and when these applies system evolves to a quality solution quickly. And the end of this process a new population will be formed.

Chapter 4: Implementation

Having presented a research methodology in Chapter 3 it is important to verify its acceptance through an implementation. This chapter provides implementation specific information that includes languages, platforms, and frameworks used and the rationale behind each choice. Furthermore, implementation chapter highlights the coding best practises followed and important algorithms used to achieve expected functionality and behaviour. Furthermore, the scope of this implementation limited to a prototype, but included all the important features highlighted in previous chapters.

4.1 Languages, platforms, and frameworks used

As the operation system for this prototype, Microsoft Windows 10 Home is selected mainly due to the familiarity with operating system and its easiness on setting up the environment and configurations. As system hardware: Intel i3-4030U CPU at 1.99GHz, 6Gb RAM, 250Gb Hard Disk drive, and intel inbuilt graphic chip were used.

Oracle Java[1]is used as the programming language mainly due to its core features:

- Java is platform-independent: Java is well known for its ability of running on any operating system when given a Java implementation (which are not linked to any operating system specific libraries). Therefore, having required Java Runtime Environment (JRE) on any platform it is easy to execute a compiled java program. This feature very important as I have selected MS Windows as my working operating system, but once this is implemented this should execute on any popular platform which are using in the industry.
- Java is object-oriented: Object oriented approach provides to create modular programs and reusable codes. Therefore, complex problems can be decomposed into manageable modules and focus can be given to a specific part independent from the rest of complexity. Furthermore, this provides to have reusable codes and therefore cleaner and less error prone code can be developed. In additionally by following object oriented concepts and proper design patterns, implementations are more robust and less error prone.
- Java is easy to learn: Java is one of the very famous programming language among many programmers and industries. The main reason for this is, Java was designed to be easy to use. To achieve that it should be easy to write, compile, debug, and learn

[1]https://www.oracle.com/java/index.html

Java, and they have facilitated all the features to achieve this. As I am lacking with the programming experience, Java is the best choice for me, as it is very easy and there are plenty of resources, tutorials, and community support if there is a help needed.

- Java is robust: Java is a compiled programming language; therefore, it checks all the errors (except runtime errors) at the time of compiling. Robust means reliability, and therefore, having successfully compiled Java program, there is a very high confident that there are no syntactical errors. This is very helpful, as I do not need to worry on any syntactical errors and its ability to pointing where the errors are.
- Java is multithreaded: Multithreaded is the capability for a program to perform several tasks simultaneously within a program. This is very useful specially in optimization problems. As explained in Chapter 2 SPS is an optimisation problem and certain parts of the process can be performed simultaneously, mainly to reduce the huge time which is required in solving a complex optimization problem. Nevertheless, though this feature is available with Java, I am not intending to use it with my current skills in Java programming. Nevertheless, this could be a good option for a future work.

As Integrated Development Environment (IDE) both Eclipse[2] and NetBeans[3] were used. Both are specific for Java platforms and provides very easy environment to write and configure java applications. NetBeans specifically used for the GUI design as it is reach with designing a GUI with drag and dropping components. Eclipse mainly used as the IDE for GA engine as it is relatively use less memory (my computer is not having a large memory capacity). Furthermore, SQLite[4]used as the database, mainly to store employee and task data. SQLite is a library that implements a self-contained, server less, zero-configuration, transactional SQL database engine. This database is very light and easy to implement, design a database and no configurations are required.

Having a database to store input data, still it is found that it will be hard to visualise and edit a large software project data. Therefore, the data stored in SQLite was extracted to an XML file that used as the input to the GA. This has few benefits, first: it is easy to validate consistency and structure of the data, and secondly most project managers in Software Industry more comfortable with editing an XML file for a large project (this save them lot of time and they can very easily navigate whether they want in the data storage in XML form).

[2]https://www.eclipse.org/ide/
[3]https://netbeans.org
[4]https://www.sqlite.org

To handle the XML data with Java, XStream[5]is used. XStream is a simple library to serialize objects to XML and back again, and therefore; I could easily read the XML data and populate necessary class objects with those. This library is open-source, easy to use, has good tutorials, and community support.

To develop Graphical User Interfaces (GUIs) in Java few java packages were used mainly Swing and AWT. With the NetBeans IDE, designing and developing Java GUIs were very easy and it was drag and drop the necessary components on appropriate layout. Furthermore, as my system is a standalone application, I did not explore any web technologies, though that is also an option for future work. Plotting a Gant Chart for identified software project schedule was another key requirement in this work and it was bit difficult to achieve merely through NetBeans with AWT and Swing. For this reason, many open source java chart libraries were explored. It was found that JFreeChart[6] is one library that makes it easy for developers to display professional quality charts in their application. Also this library equipped with well-documented API, multiple types of chart support, easy to extend, support many output types, and open-source. These reasons were good enough to select JFreeChart as the library for polling charts with community support and tutorials available on the web.

4.2 GA specific implementation

The core of this research is to use GA for SPS. Therefore, main focus goes to this and it was challenging. There were many toolkits and libraries for GA specific implementations. Among them for Java platform: Jenetics[7], codeproject Genetic Algorithm Library[8], GAUL[9], and ECJ[10] are interesting toolkits and libraries. I found though these libraries are interesting and useful; they need relatively more programming experience and knowledge to use in applications. Therefore, I stick to basic GA requirements and developed a simple GA engine/framework specific to SPS under the scope of this research.

For this mainly 4 packages were identified, namely 1) com.ucsc.mit.model, 2) com.ucsc.mit.ga, 3) com.ucsc.mit.tool, and 4) com.ucsc.mit.gui. Hear the package of gui includes all the UI related classes for outputs. The package of model represents necessary class structure to represent SPS specific data which getting as inputs from an external XML file (XML file will be populated by the data that has stored in a DB). This model package

[5]http://x-stream.github.io
[6]http://www.jfree.org/jfreechart/
[7]http://jenetics.io
[8]https://www.codeproject.com/Articles/26203/Genetic-Algorithm-Library
[9]http://gaul.sourceforge.net
[10]https://cs.gmu.edu/~eclab/projects/ecj/

mainly includes java classes: Schedule.java, Employee.java, Task.java, Skill.java, PrecedenceGraph.java, AdjacentVertice.java, and Field.java. These classes contribute to form a necessary data structure to holds SPS related information.

Ga package includes all the necessary support for GA specific needs. This package consists with classes: GAProcess.java, Population.java, Chromosome.java, Gene.java, FitnessCalculation.java, TournamentSelection.java,DataModel.java, GantChatData.java, and TimeInterval.java. These classes individually represent key concepts in GA and also supportive classes needed in SPS under GA. GAProcess class provides the overall controller in GA process, while Population, Chromosome, and Gene classes represents population of individuals, SPS individual solution/candidate, and single atomic unit in an individual (an individual is a collection of genes) respectively. Furthermore, TournamentSelection includes the functionality presented in Section 3.3.5 that leads to select a suitable chromosome (a solution) form a population. FitnessCalculation main purpose is to evaluate the quality of a given chromosome and quantify its quality as presented in Section 3.3.3. When calculating the fitness value FitnessCalculation uses the support of classes GantChatData, and TimeInterval. GantChatData populate necessary data to draw a Gantt chart when given a chromosome and TimeInterval class use as a data structure of time specific information for each task. DataModel class stores an internal data structure representation of Schedule class in model package, and this class extracts all the properties out from the Schedule such that those can be accessed by any class in the GA package. DataModel class act as an instance of singleton design pattern.

Tool package holds the main controller of the system and to store the constants values generic to the whole system. Therefore, two class holds hear namely: IntelligentProjectScheduler.java and ConfigurationData.java. IntelligentProjectScheduler is the main Java class with the run method and aggregates the logic as necessary in internally. ConfigurationData provides all the parameter values required for the project.

4.3 Algorithms and Data Structures

For this implementation it was required to use many data structures available in Java and also to implement algorithms. Mainly as data structures HashMap, List, Set, and ArrayList were used. Having these data structures, it was easy to interact with data with various implementation choices made. Furthermore, there were many specific data structures also implemented aggregating some of the classes highlighted in Section 4.2 with above mentioned data structures.

Another set of challenges in this implementation was to identify necessary algorithms required for behaviours in methods. Among them first it was required to have a method that provides who are the employees for a given task. This was very important in many other algorithms also and helped to reduce the searching time a lot under various steps.

```java
/**
* This method extracts individual employees per each skill.
*/
private void whoHasSkillX() {
        List<Employee> employees = model.getEmployees();
        for (Employee emp : employees) {
                String employeeID = emp.getEmployeeID();
                List<Skill> employeeSkills = emp.getEmployeeSkills();
                Set<Skill> employeeSkillsSet = newHashSet<>(employeeSkills);
                for (Skill empSkill : employeeSkillsSet) {
                        String skillName = empSkill.getSkillName();
                        if (whoHasGivenSkill.containsKey(skillName)) {
                                whoHasGivenSkill.get(skillName).add(employeeID);
                        } else {
                                whoHasGivenSkill.put(skillName, newHashSet<>());
                                whoHasGivenSkill.get(skillName).add(employeeID);
                        }
                }
        }
}
```

Another task was to assign employees randomly to each tasks. This was required in forming the initial population. For this a simple random function used to select employees randomly.

```java
/**
* This method select employees that has the skill for a given task. In this
 * regard first it finds for each skill who has the skill among employees.
 * Having that employee list it randomly pick one employee. This continues
 * to all the skills required and finally randomly selected employee set
 * will passed
 *
 * @paramemployeeSkills:
 *             List of Skill
 * @return A set of employee IDs
 */
public HashSet<String> getRandomEmployeesForTask(List<Skill>employeeSkills) {
        HashSet<String> randomEmpSet = newHashSet<String>();
        for (Skill skil : employeeSkills) {
                HashSet<String> empSet = whoHasGivenSkill.get(skil.getSkillName());
                intmax = empSet.size() - 1;
                intmin = 0;
                intpointer = random.nextInt((max - min) + 1) + min;
                String empID = (String) empSet.toArray()[pointer];
                randomEmpSet.add(empID);
        }
        return randomEmpSet;
}
```

Furthermore, fitness calculations need suitable algorithms to calculate the necessary values as in Section 3.3.3. For these purpose mainly 5 methods were identified. Among them the first is to calculate the project cost. For this salaries of individuals who are assigned to a given project was calculated separately and getting the summation of all such individuals assumed to be the main cost for the project. It was calculated as per the bellow code segment.

```
private double calculateProjectCost(Gene[][] solution) {
        double totalProjectCost = 0;
        List<Employee> employees = dModel.getModel().getEmployees();
        for (Employee emp : employees) {
                String empID = emp.getEmployeeID();
                float salary = emp.getEmployeeSalary();
                int empRowNum = dModel.getEmployeeOrderForArry().get(empID);
                double thisTaskCost = 0.0;
                List<Task> tasks = dModel.getModel().getTasks();
                for (Task task : tasks) {
                        String taskID = task.getTaskID();
                        int taskColumnNum = dModel.getTaskOrderForArry().get(taskID);
                        String geneValInString =
                        solution[empRowNum][taskColumnNum].getDataInString();
                        float empDedOnTask =
dModel.getEmpDedicationFrom4DigitBinaryEncodingToReal().get(geneValInString);
                        double taskDuration = calculateTaskDuration(solution, task);
                        thisTaskCost += (salary * empDedOnTask * taskDuration);
                }
                totalProjectCost += thisTaskCost;
        }
        return totalProjectCost;
}
```

Second calculation was the project duration and project duration is summation of task durations. Nevertheless, some of tasks can be execute in parallel and for certain tasks it may needs to wait certain tasks to be finished as per the given dependency graph information. Therefore, for this reason this is complicated to calculate and needs to separately combined duration of individual tasks together with dependency graph information. For this following code segment was used.

```
private double calculateTaskDuration (Gene[][] solution, Task task) {
        double taskDuration = 0;
        String taskID = task.getTaskID();
        float taskEffort = task.getTaskEffort();
        int taskColumnNum = dModel.getTaskOrderForArry().get(taskID);
        double EmpsTotDedicationOnTask = 0.0;
        for (inti = 0; i<dModel.getNumOfEmp(); i++) {
                String geneValInString =
                solution[i][taskColumnNum].getDataInString();
                EmpsTotDedicationOnTask +=
dModel.getEmpDedicationFrom4DigitBinaryEncodingToReal().get(geneValInString);
        }
        if (EmpsTotDedicationOnTask> 0) {
                taskDuration = taskEffort / EmpsTotDedicationOnTask;
        }
        return taskDuration;
}
```

Another calculation was to count number of employees who have assigned to a tasks but who doesn’t have at least one skill which is required for that specific task. For this calculation a specific code snippet was used that first iterate each task and extracted the skill required. Having knowing the skill required for a task, then it is matter of finding the skills of each employee assigned to that task. If at least there is one skill with that employee that is equal to the required skills in the task it is a passing condition. Otherwise a counter was increased by one.

```
private int calculateProjectSkillsMissing(Gene[][] solution) {
        int counter = 0;
        List<Task> tasks = dModel.getModel().getTasks();
        for (Task task : tasks) {
                List<Skill> requiredSkillsInTask = task.getEmployeeSkills();
                Set<String> requiredSkillsInNamesInTask =
                        getskillNameSet(requiredSkillsInTask);
                String taskID = task.getTaskID();
                int taskColumnNum = dModel.getTaskOrderForArry().get(taskID);
                List<Employee> employees = dModel.getModel().getEmployees();
                Set<String> skillsInNamesInEmps = newHashSet<String>();
                for (Employee emp : employees) {
                        String empID = emp.getEmployeeID();
                        int empRowNum = dModel.getEmployeeOrderForArry().get(empID);
                        String geneValInString =
                                solution[empRowNum][taskColumnNum].getDataInString();
                        if (!geneValInString.equals("0000")) {
                                List<Skill> empSkills = emp.getEmployeeSkills();
                                skillsInNamesInEmps.addAll(getskillNameSet(empSkills));
                        }
                }
                if (!skillsInNamesInEmps.containsAll(requiredSkillsInNamesInTask)) {
                        counter++;
                }
        }
        return counter;
}
```

The other important calculation was the project overwork cost. For this it was required to calculate the overwork time of each employee separately. This was achieved through by calculating the multiple dedications of each employee if they are working on more than one task parallel. Therefore, it was required to identify at each time point parallel tasks each employee is working on and then to check whether that is exceeding the max dedication of that employee. If that’s the case, then such cumulative values lead to over work value of that employee.

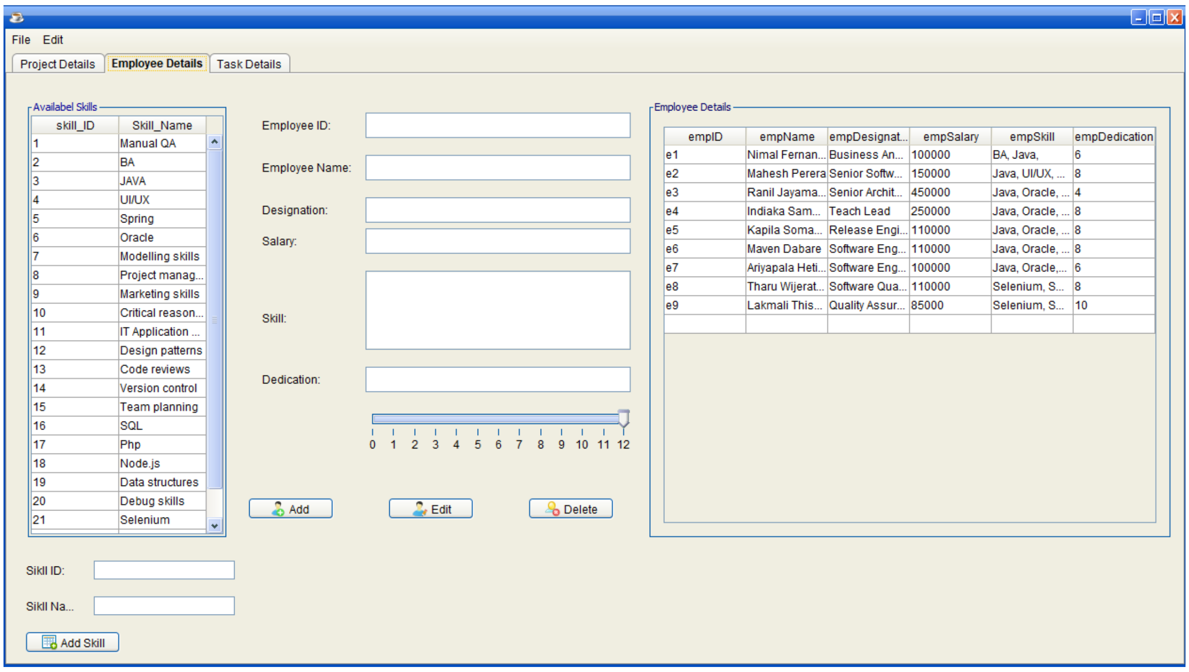
```
private double overWorkOfEmployeeX(Gene[][] solution, HashMap<String, TimeInterval> gantChatData,
String empID) {
int empRowNum = dModel.getEmployeeOrderForArry().get(empID);
HashMap<String, Integer> taskOrder = dModel.getTaskOrderForArry();
Set<String> visitedTasks = new HashSet<String>();
double overTime = 0;
for (String taskID : taskOrder.keySet()) {
        int taskColumnNum = taskOrder.get(taskID);
        visitedTasks.add(taskID);
        if (!solution[empRowNum][taskColumnNum].getDataInString().equals("0000")) {
        for (String key : taskOrder.keySet()) {
                int temp = taskOrder.get(key);
                if (!solution[empRowNum][temp].getDataInString().equals("0000")) {
                if (!visitedTasks.contains(key)) {
                        double startTimeRef = gantChatData.get(taskID).getStartTime();
                        double endTimeRef = gantChatData.get(taskID).getEndTime();
                        double startTimeCur =gantChatData.get(key).getStartTime();
                        double endTimeCur = gantChatData.get(key).getEndTime();
                        if((startTimeRef<= startTimeCur) &&((endTimeRef - startTimeCur) > 0)){
                        double dedicationGap = calcualteExtraDedicationOnOverWork(solution,
empID, empRowNum,taskColumnNum, temp);
                        if (dedicationGap> 0) {
                                overTime += (dedicationGap * (endTimeRef - startTimeCur));
                        }
                        } else if ((startTimeCur<startTimeRef) &&((endTimeCur - startTimeRef) > 0)){
                        double dedicationGap = calcualteExtraDedicationOnOverWork(solution,
                                        empID, empRowNum,taskColumnNum, temp);
                        if (dedicationGap> 0) {
                                overTime += (dedicationGap * (endTimeCur - startTimeRef));
                        }
                }
                }
        }
        }
}
visitedTasks.add(taskID);
}
return overTime;
}
```

4.4 User Interfaces and XML data

To facilitate the system functionality a simple graphical user interface also developed that integrated with the GA process. In this regards mainly two forms are developed to get user inputs through GUI and another interface to visualise the output (which is the proposed schedule for a software project). The below UIs are used to add the employee details and the task details for the particular project. These interfaces provide necessary functionality to add employee and task data, nevertheless, in practical context for a large project this is difficult to purely work on these type of interfaces. Therefore, most of project managers are interesting to have both GUIs and XML files that includes the same data. A data on a XML file can be edited much easily and they can quickly search necessary data and do the needful. Therefore, an XML file also introduced with the same data that will be passed to the GA algorithm. The schedule details interface helps to generate xml file and make the schedule. It generate the Gantt chart and feasibleness of the solution. And also provide project schedule fitness data with the solution matrix . The schedule fitness data provided below.

- Total Project Cost
- Total Project Duration
- Number of tasks with no employees
- Number of Tasks with Missing Skills,
- Total Overwork Value,
- Current Value for feasibleness (1 / (q + p))

A GUI is designed for handling employee data, which used to add, edit, and delete and employee or employee data. Furthermore, this provides a full view of all the employee details, so that this can be used to easily select and modify or delete any employee. Also there is another option given (left side of the employee details page) to add any skills that needs to maintain for the company. Figure 4-1 provides a screen capture of employee details.

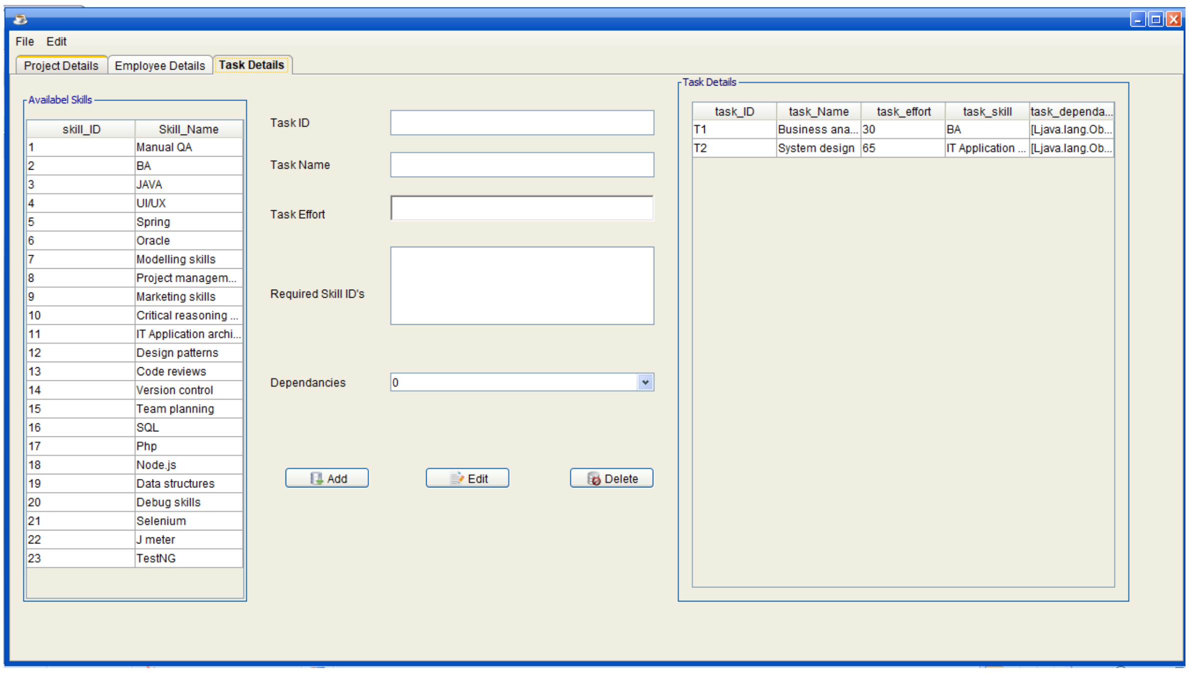


skill_ID	Skill_Name
1	Manual QA
2	BA
3	JAVA
4	UI/UX
5	Spring
6	Oracle
7	Modelling skills
8	Project manag...
9	Marketing skills
10	Critical reason...
11	IT Application ...
12	Design patterns
13	Code reviews
14	Version control
15	Team planning
16	SQL
17	Php
18	Node.js
19	Data structures
20	Debug skills
21	Selenium

empID	empName	empDesignat...	empSalary	empSkill	empDedication
e1	Nimal Fernan...	Business An...	100000	BA, Java,	6
e2	Mahesh Perera	Senior Softw...	150000	Java, UI/UX, ...	8
e3	Ranil Jayama...	Senior Archit...	450000	Java, Oracle, ...	4
e4	Indiaka Sam...	Teach Lead	250000	Java, Oracle, ...	8
e5	Kapila Soma...	Release Engi...	110000	Java, Oracle, ...	8
e6	Maven Dabare	Software Eng...	110000	Java, Oracle, ...	8
e7	Ariyapala Heti...	Software Eng...	100000	Java, Oracle,...	6
e8	Tharu Wijerat...	Software Qua...	110000	Selenium, S...	8
e9	Lakmali This...	Quality Assur...	85000	Selenium, S...	10

Figure 4-1: GUI of employee details

Another GUI is designed and developed for task details. This includes all the task related information and provides the option to add, edit, or delete and task in the system. Furthermore, a full view of all the task details also provided, therefore, a use can select any task data to either edit or delete. Figure 4-2 provides a screen caption of task details page.

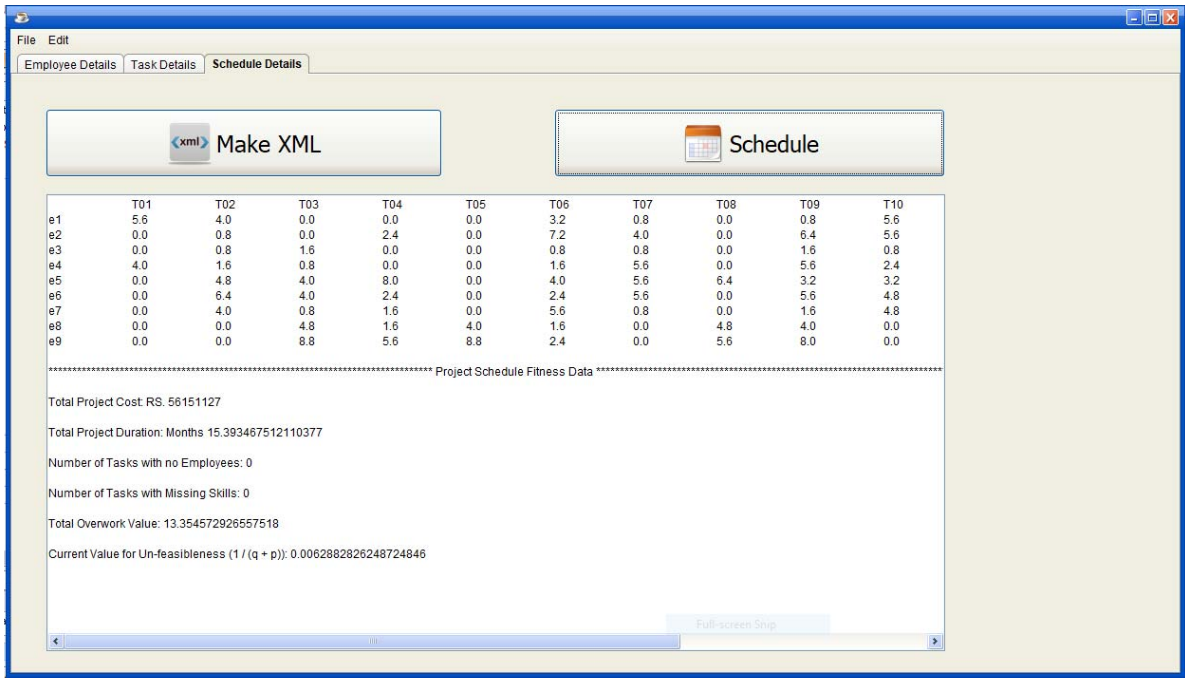


skill_ID	Skill_Name
1	Manual QA
2	BA
3	JAVA
4	UI/UX
5	Spring
6	Oracle
7	Modelling skills
8	Project managem...
9	Marketing skills
10	Critical reasoning ...
11	IT Application archi...
12	Design patterns
13	Code reviews
14	Version control
15	Team planning
16	SQL
17	Php
18	Node.js
19	Data structures
20	Debug skills
21	Selenium
22	J meter
23	TestNG

task_ID	task_Name	task_effort	task_skill	task_dependa...
T1	Business ana...	30	BA	[Ljava.lang.Ob...
T2	System design	65	IT Application ...	[Ljava.lang.Ob...

Figure 4-2: GUI of task details

Figure 4-3, provides the functionality necessary to GA process. Having an option to insert the data through interface, this may not feasible for large software project. Therefore GA process input, XML file with all the required data used as the input. This page include two buttons where 'Make XML' is to generate the XML file from the data entered from other pages, where as having generated or manually updated a XML file (as per the format required) 'Schedule' button generate the SPS as per the GA process developed. Under these two buttons, the summary output of final solution generated were included.



```
       T01    T02    T03    T04    T05    T06    T07    T08    T09    T10
e1     5.6    4.0    0.0    0.0    0.0    3.2    0.8    0.0    0.8    5.6
e2     0.0    0.8    0.0    2.4    0.0    7.2    4.0    0.0    6.4    5.6
e3     0.0    0.8    1.6    0.0    0.0    0.8    0.8    0.0    1.6    0.8
e4     4.0    1.6    0.8    0.0    0.0    1.6    5.6    0.0    5.6    2.4
e5     0.0    4.8    4.0    8.0    0.0    4.0    5.6    6.4    3.2    3.2
e6     0.0    6.4    4.0    2.4    0.0    2.4    5.6    0.0    5.6    4.8
e7     0.0    4.0    0.8    1.6    0.0    5.6    0.8    0.0    1.6    4.8
e8     0.0    0.0    4.8    1.6    4.0    1.6    0.0    4.8    4.0    0.0
e9     0.0    0.0    8.8    5.6    8.8    2.4    0.0    5.6    8.0    0.0

****************************************************************************** Project Schedule Fitness Data ******************************************************************************

Total Project Cost: RS. 56151127

Total Project Duration: Months 15.393467512110377

Number of Tasks with no Employees: 0

Number of Tasks with Missing Skills: 0

Total Overwork Value: 13.354572926557518

Current Value for Un-feasibleness (1 / (q + p)): 0.0062882826248724846
```

Figure 4-3: Project schedule fitness data and solution matrix

The schedule generated from GA approach is visualised as a Gantt Chart. This Chart includes the time that will be taken for each task with adhering to the precedence data provided (the tasks which cannot execute in parallel as there are dependencies among them). Figure 4-4 provides a screen capture of a schedule generated by the system. Figure 4-5 provides the feasible development (1/ (p + q)) of the process through each evolution cycle.

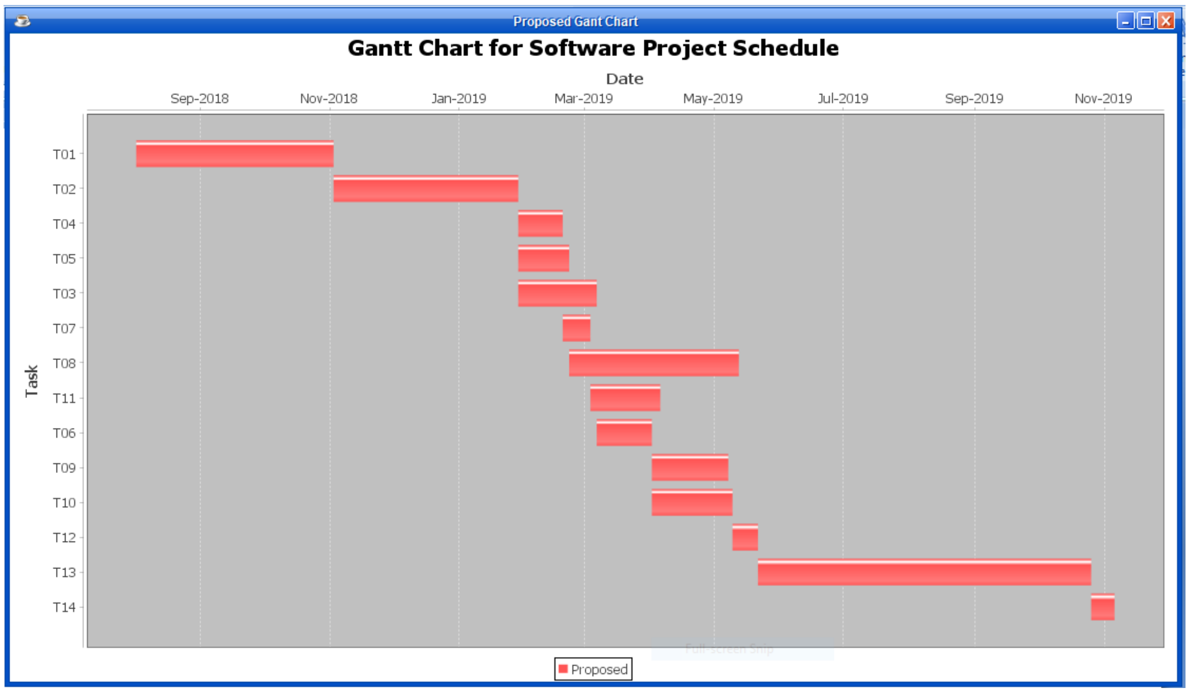


Figure 4-4: Gantt Chart for the project schedule

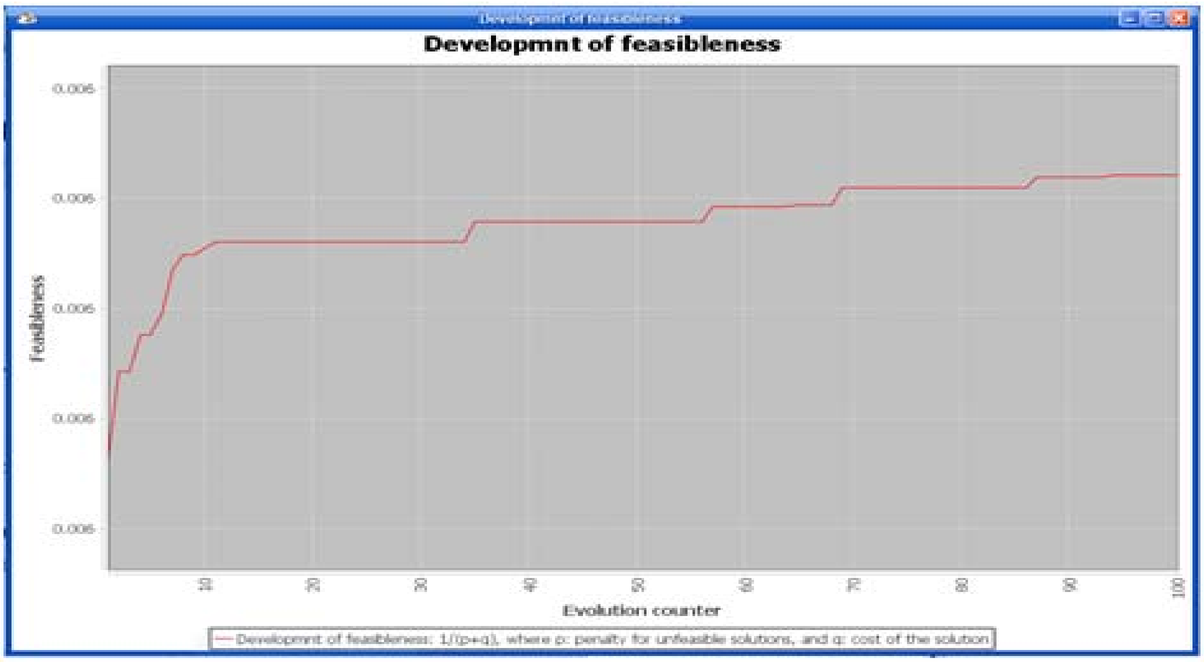


Figure 4-5: Feasibleness graph

Chapter 5: Evaluation and Results

Evaluation of a research is crucial as it provides the confidence about the proposed approach. For this it is essential to identify how much the original intended goals had been achieved with the results obtained. This applies to this research as well and this evaluation can be separated into two. First there has to have a software project evaluation specifically on colleting the data from a user through developed GUIs. Secondly, it is important to validate the quality and correctness of the generated SPS by the GA based approach implemented. From these two, second carries more weight as this being a research project.

For all of these it is necessary to have a sample project data that can be used in this evaluation approach. As a fact that a project data was identified which is presented in Table 5-1 and 5-2. In hear we need to get these data from the user to initiate the SPS system. This includes tasks data and employee data. Table 5-1 provides the task specific data that includes task id, task name, effort on task required, and skills required for that specific task. Furthermore, Table 5-2 provides employee data with their skills. For this it is considered 14 tasks and 9 employees.

Table 5-1: Task data and task dependencies

Task #	Task	Task effort	Skill Required
T1	Business analysis	30	BA,
T2	System design	65	IT Application architecture, Java,
T3	DB Implementation	30	Oracle, SQL
T4	UI Design	15	UI/UX, SQL, Node.js,
T5	Create Test Case	10	Manual QA,
T6	Core Implantation	25	Java, Oracle, SQL
T7	User Interface Review	10	UI/UX, BA, Java
T8	Create Automation Framework	45	Selenium, SQL, J meter, TestNG
T9	Implement Feature 1	45	Java, SQL, Oracle, Data structures
T10	Implement Feature 2	35	Oracle, Java,
T11	UI Implantation	45	UI/UX, SQL, Java, Node.js
T12	System Integration	10	Java, SQL, Oracle, Node.js
T13	Integration Testing	75	Selenium, SQL, TestNG, Manual QA, SQL, J meter
T14	UAT	5	BA

Table 5-2: Employee data

ID	Employee name	Designation	Salary (LKR)	Skills	Dedication
e1	Nimal Fernando	Business Analysist	100000	BA, Java,	6
e2	Mahesh Perera	Senior Software Engineer UI	150000	Java, UI/UX, Spring	8
e3	Ranil Jayamaha	Senior Architect	450000	Java, Oracle, Spring, Modeling skills, Project management skills, Marketing skills, Critical reasoning skills, IT Application architecture	4
e4	IndiakaSamarasinghe	Teach Lead	250000	Java, Oracle, Spring, Design patterns, Code reviews, Version control, Team planning	8
e5	Kapila Somabandu	Release Engineer	110000	Java, Oracle, SQL, Php, Node.js, Data structures, Debug skills	8
e6	Maven Dabare	Software Engineer	110000	Java, Oracle, Node.js, Data structures	8
e7	Ariyapala Hetiarachi	Software Engineer	100000	Java, Oracle,Php, Node.js	6
e8	Tharu Wijerathne	Software Quality Assurance Engineer	110000	Selenium, SQL, J meter, TestNG, Manual QA	8
e9	LakmaliThisera	Quality Assurance Engineer	85000	Selenium, SQL, Manual QA	10

It is not enough to have the data in Table 5-1 and 5-2. In a software project, tasks include precedence and some tasks cannot perform in parallel. For example, system testing should starts only if the unit tests are successfully completed. For this requirement, task dependency graph can be used. A precedence graph provides the details about which tasks must be completed before a new task is begun. Precedence graph is an acyclic directed graph and Figure 5-1 presents this for the proposed data set.

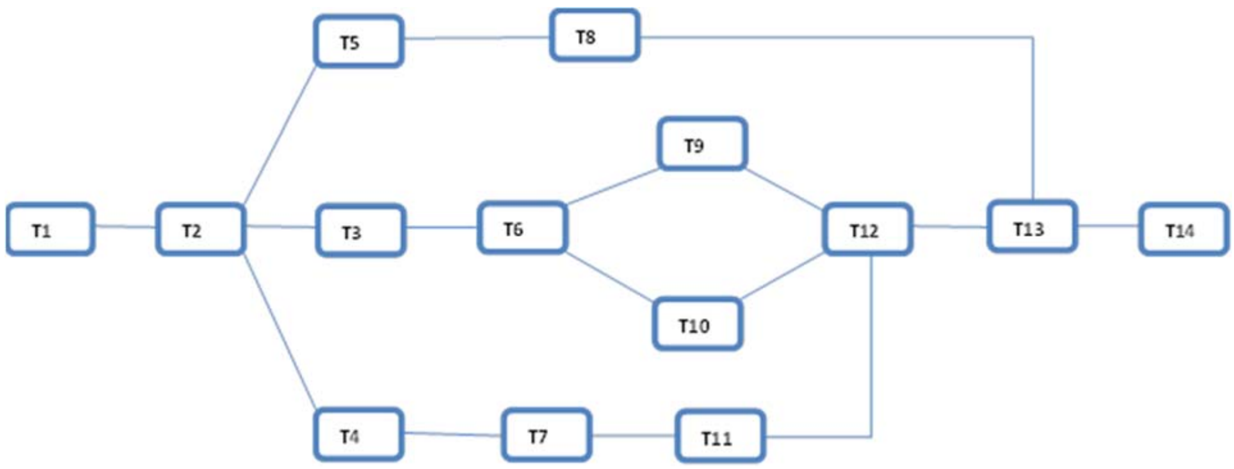


Figure 5-1: Precedence graph for data in Table 5-1

5.1 Test cases for user interfaces

A test case is a set of test inputs, execution conditions, and expected results developed for a particular objective. System functionality could be tested with properly planned test cases. The system was decomposed into modules to reduce the complexity and to minimize the dependency. Functional test cases executed to check the basic functionality of the employee UI and task UI. Test cases of major modules are presented below. Main functionalities of employee UI includes:

- Add employee
- Edit employee
- Delete Employee
- Add employee skill

Test cases and expected results on related functionalities on employees can be found in appendix A (Table A-1: Employee information related test cases).

Functionalities of the task details GUI includes 1) Add task, 2) Edit task, and 3) Delete task. To cover this functionality a test plan is presented in appendix A (Table A-2: Task information related test cases).

5.2 Research scope for the evaluation

Section 1.3 provides the scope of this project and it applies for the evaluation too. The scope of this project mainly condensed to a working demonstrable prototype that has the potential to provide a rational intelligent project schedule when given necessary information. In practical context it is computationally a difficult task to find the best project schedule for a complex system and it may take a lengthy time to derive a solution. Therefore, in the scope of this project the main expectation is to obtain a rational project schedule than the best project schedule. Therefore, having this feature, the proposed system has the ability to stop the searching on a large search space once it is found an optimal solution which can be logically justifiable.

In addition to the rationality system is expected to be intelligent as well. In this context, what is intelligent means that is the ability to find a good solution as an autonomous process where there is no human intervention. Nevertheless, at the beginning a user should need to provide the necessary data and constraints of the software project as an input as presented in Section 5.1. Furthermore, finding the right parameter values will not be considered as the part of the intelligence of the system.

This project is considered to be as a research project than a software development IT project. The main reason why this is called as a research project is that there is no specific approach that always provides a rational project schedule intelligently. AI related techniques have been used for this purpose though it is difficult to identify a suitable approach considering all the specific features required.

5.2.1 Assumptions

Following are identified as assumptions on this research:

- Given project constraints won't be change in the middle of the project, or if it is changed, the schedule should be regenerated.
- It is assumed that data which will be provided as inputs are correct and accurate.
- Standard computer hardware (processing power and RAM capacity) available are sufficient enough to execute GA based application.
- Intelligence can be measured through comparing the quality of an output generated by system and a human.
- It is not required to generate project schedules for multiple projects in a single execution.

5.3 Turing Test Approach

As highlighted in Section 1.2 the main research question is how to generate a project schedule for a complex software project as an automated task. This being an intelligent application its validity is difficult to quantify purely through traditional testing approaches. Turing test approach is common for validating the acceptance of most of the intelligent applications. Turing test approach the main rational is to compare the results of an intelligent application with a real human being. As we always believe that humans (with necessary knowledge) are intelligent, if it is not distinguishable the results given by a real human expert with the same from an intelligent application, it is acceptable to conclude that application is also intelligent.

More formally, the Turing Test, proposed by Alan Turing (Turing, 1950), was designed to provide a satisfactory operational definition of intelligence. Turing defined intelligent behaviour as the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator. Roughly speaking, the test he proposed is that the computer should be interrogated by a human via a teletype, and passes the test if the interrogator cannot tell if there is a computer or a human at the other end.

This specific approach will be used as the main testing strategy for evaluating the SPS approach proposed with GA technology. In this regard it is possible to consider the factors that will be considered by a real human expert whether a given SPS is a good quality solution. Such factors include:

- Total cost of a solution
- Total project duration of a solution
- Total number of tasks with no employee appointed
- The number of skills missing in order to perform all project tasks
- The total overwork of the project

Having these factors, it is easily possible to quantify the quality of solutions provided by the implemented approach. This provides good indication how closer the implemented approach from a real human expert who is performing SPS task.

For this reason, a questionnaire is developed that is available at below URL and shared with 10 people who are responsible for developing software project schedules (these questions will be listed down in Section 5.3.1):

https://docs.google.com/forms/d/1dPAstvFmEU3tVd_AKhr3moIMN3H8QzxEv4TWeaA5Wyc/viewform?edit_requested=true

This questionnaire, the use case explained in this chapter was used and provided all the information. Furthermore, this includes a solution subjected by the developed prototype. Nevertheless, in this questionnaire, it was not mentioned that the provided schedule is from a system, but asked from them to compare a schedule that they proposed (for the given data) to compare the quality of provided result. This strategy goes with the Turing approach and provides good information to validate the quality of proposed approach. Questions of this questionnaire includes in Appendix B.

5.3.1 Analysis of data gathered from questionnaire

The results gathered from Turing Test approach provide a good intuition for how this approach useful for industry people. This questionnaire was filled by 10 experts (6 from associate PM to PM positions, and 4 Tech Leads).

The first question was: "How long did you take to finish your schedule". The reason behind asking this question was to find how much time it takes for an expert to create a project schedule from given data. From the developed prototype for this, it took only less than 2 hours. Nevertheless, as per the results in Figure 5-2, for 50% it has taken around 45mins to complete this task manually. This provides good qualitative result on how fast the proposed approach and the time that it will save on complex scheduling problems. Especially when it comes to complex projects, there are many tasks and dependencies needs to be considered and sometimes more than one project manager is required for this and even take few weeks to complete. In this regard it is clear that receptively the proposed approach is fast.

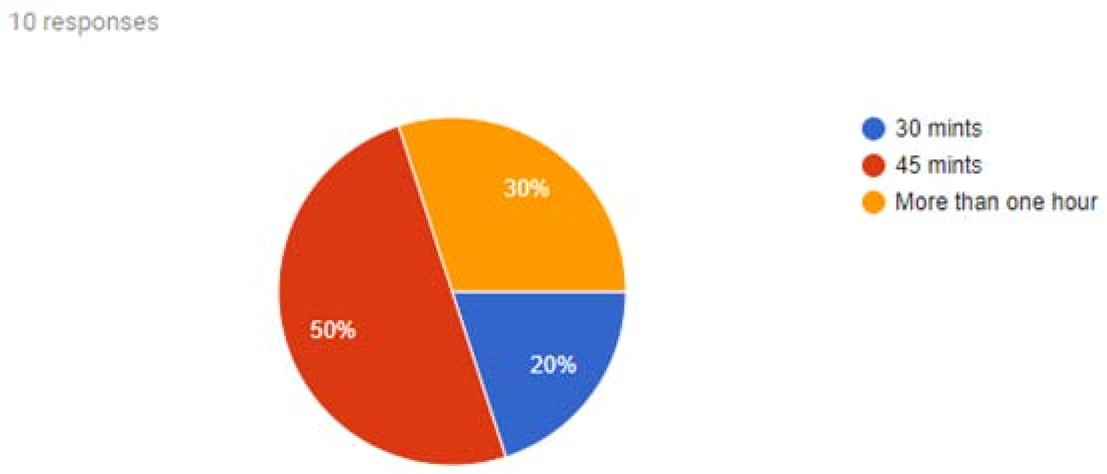


Figure 5-2: Results for the question: how long did you take to finish your schedule

The second question is: "How easiest to generate a new different schedule for the same data set". This question is very important due to the dynamic nature in software projects. Most of time people are bias with what they found; nevertheless, there may be alternatives some times better than what we found. In the proposed approach with GA, it is always easy to have multiple solutions nevertheless, as per the experts answers it is clear than for 60% to have an alternative option is not easy (see Figure 5-3). This means if they need in average 45min to have the first solution, most probably they need more than that to find the second solution. In this regards the proposed approach is a success, as it can provide multiple solutions with most of the time within the same time interval.

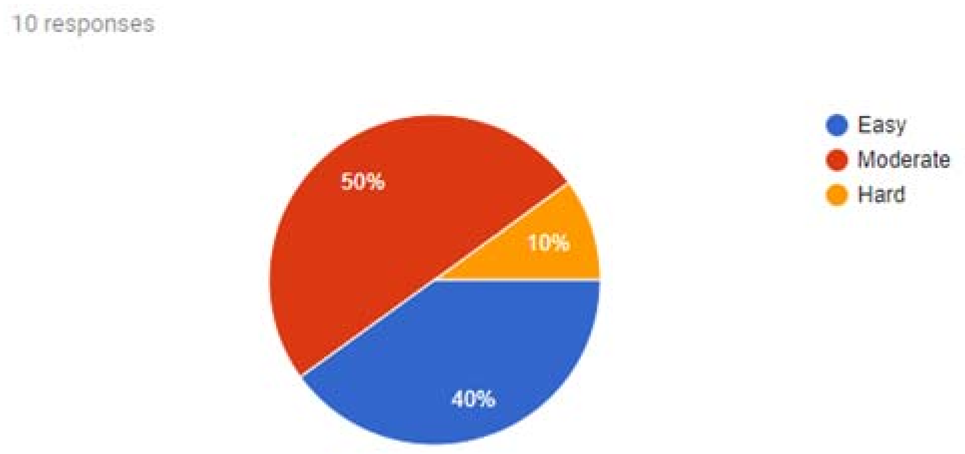


Figure 5-3: Results for the question: How easiest to generate a new different schedule for the same data set

The next question was: "How better the project duration in what you have drawn manually relative to the given reference project schedule. Manually drawn schedule duration = t1 Reference schedule duration = t2". This is a very important question specific to SPS. Always it is important to finish project sooner (together with minimising the cost). In this regard, this question provides good idea to evaluate the time aspect of the proposed approach. As per the Figure 5-4, it is clear that 60% have found that their solution provides better time than what approach provided. This clearly implies that the proposed approach is not adequately optimised and it needs further fine tuning.

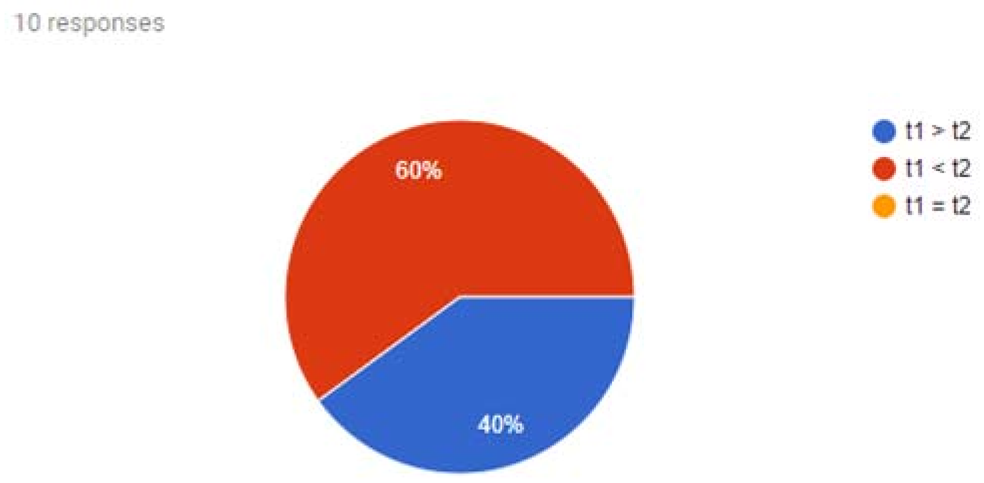


Figure 5-4: Results for the question: The time aspect of the solution

The fourth question: "Total project cost also mainly assumed as the salaries of employees. So how Better the project cost in what you have drawn manually relative to the given reference project schedule. Manually drawn schedule cost = c1 Reference schedule cost = c2", provides good information about the total cost of the solution. In every project similar to the time, it is very important the cost of the project as well. As per the Figure 5-5, 50%

have found that the total cost of given solution is better than what they found. Nevertheless, the 50% have found that their approach is cheaper than given reference. When considering both these facts it is clear that current approach does not provide the most economical solution, but still it is more or less close to the average. Therefore, with further improvements, this factor can be further reduced.

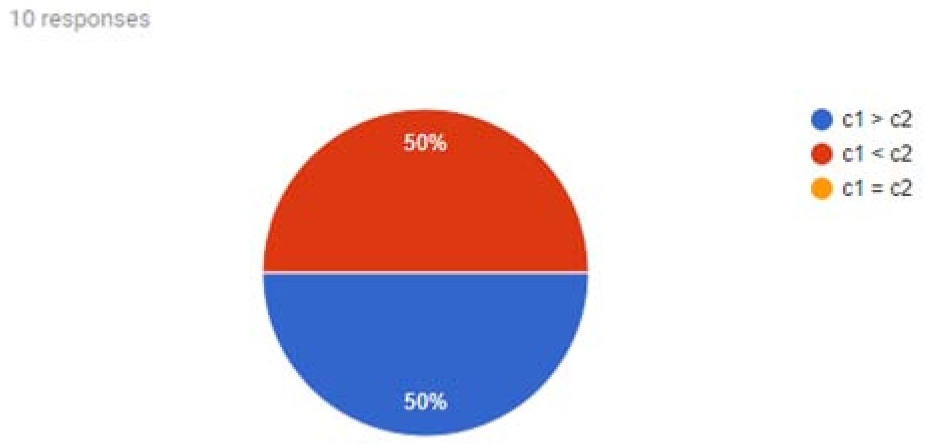


Figure 5-5: Results for the question: The total cost of the project

The next question is: “Calculate the total over work for the schedule and compare with the reference given Manually calculated over work - 01 Reference over work- O20”. This is partly related with project cost and taking extra time as overwork always a negative point in SPS. As per the Figure 5-6, 60% have found that given reference use more overwork. This is also not something appreciate and clear indication for further improvements. Nevertheless, when considering the time duration, cost, and overwork, these provides a good overall interpretation that though proposed approach provide reasonable results, still there are away from the best.

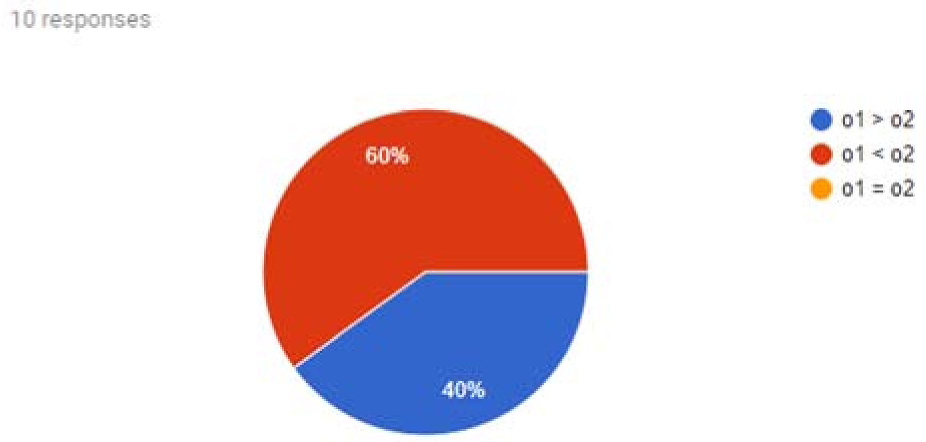


Figure 5-6: Results for the question: The total overwork info

The sixth question is: "Is there any task in your schedule with no employees assign". This question provides how complete our solution. If there is one task at least with no employees, then this clearly shows that results are absolutely pre mature and not really suitable as a solution. From, the results we received from the GA approach always at least one employee were assigned in a task. Therefore, with this approach it was not possible to find any solution where there was a task with at least one employee. When it comes to the all experts too, have found solutions that satisfies this criterion.

The next question is: "Is there any task in your schedule with missing skills". Similar to the previous question, this is also very important, as it is not good to have a task where it has assigned the employees that at least have all the skills required to complete it as a whole. The proposed approach always satisfied this criterion while one expert has provided a solution where there is one skill is missing in assigned employees for that task. Figure 5-7 provides this observation.

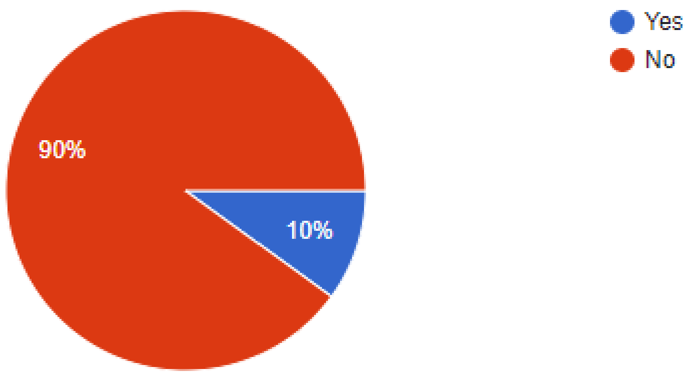


Figure 5-7: Results for the question: Tasks with missing skills

The seventh question is: "Do you think whether the software project scheduling system can be automated". From some of above results it is clear that the proposed approach has not provided the best results. Nevertheless, this may be due to a fact that this task being very difficult to automate. Therefore, this question is very important to understand the impression of experts about this. As per the Figure 5-8, 90% believed that it is harder to automate SPS problem. Therefore, this is a good result to justify the attempt made to automate this process though it is not still not closer to perfect.

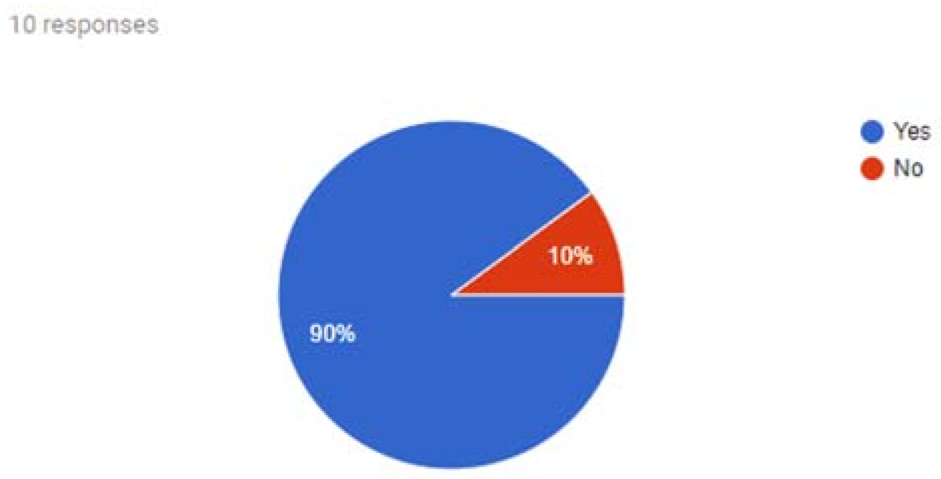


Figure 5-8: Results for the question: Do you think whether the software project scheduling system can be automated

The next two question goes with the answer for the previous question. The next two questions are: 1) If it is possible as for your experience how hard it would be to automate this problem, and 2) If there is a such a system will you be used in your project scheduling work. For the first 62.5% has answered as hard and 75% answered as yes for the next question. This further strengthen the previous argument. Even some feel that it is possible to automate SPS still most of them believe that this is a harder task, and also the most important thing is majority is willing to use if such a system exists.

With this questioner it is clear that the proposed approach is not perfect though it has some interesting properties and there is a need in the industry for better system for this purpose. The main problem in this approach is though it is providing a solution, it is still not complete enough to use as a 100% replacement for an expert.

Chapter 6: Conclusion and Future work

In this research main question was to generate a rational project schedule as an intelligent system for software project management. For this purpose, a specific approach was presented by using genetic algorithm as the key technology. This being a scheduling problem it needs to consider multiple factors when forming a rational schedule. Therefore, as features followings are used in this research (see Section 3.2):

- Employee identification
- Employee skills
- Employee Salary
- Employee dedication
- Task id
- Required skills in a task
- Required effort for a task

In the proposed approach, the first challenge was how to encode a given software project scheduling requirement into a binary representation such that it can be easily used in the GA process. A specific step was used and as stated in the Section 3.3.2 this is achieved with minimum data loosing. Having an approach to represent a problem in binary form the next step was to form an initial population. An algorithm is developed for this purpose that returns any required number of individuals with random values to use as the initial population. Having this population, the next step is to follow the GA process, where parent selection, applying genetic operations and evaluating the fitness of new individuals. For the parent selection tournament selection technique was used. Crossover and mutation were applied with appropriate probabilities (if the population size as 20–30, crossover rate as 0.75–0.95, and mutation rate as 0.005–0.01; but if the population size as50–1000, crossover rate as~0.6, and mutation rate as 0.001). In addition to crossover and mutation elicit selection applied, so that always the best candidates always moved to new populations. For the fitness evaluation complex equation (Equation 1 in Section 3.3.3) is used in which both the cost of a solution and the penalty for unfeasible solution were considered. Therefore, this became an optimisation of minimizing the both cost and penalty.

In addition to the main functionality a GUI was developed so that user can enter project specific data much easier to the system. This includes getting employee details, task details, and precedence data of the tasks. This interface provides all the basic operations on these data and user can add, delete, and edit any data as preferred. Furthermore, XML is used as the main input to the GA system. In practical context large software projects includes many employees and tasks, therefore; it is harder to deal that amount of data merely through an interface and most of the project managers are also preferred to use XML as the input data. Therefore, entered data is injected to an XML file and that was used as the main input to the system. Finally, Gantt chart is developed that is able to visualise the final rational schedule that system will be proposed.

This process is validated with proper test cases and derived interesting results. This is divided into mainly two tastings 1) GUI and 2) GA based approach. The proposed GUI passed all the test cases presented in Section 5.1. The Table A-1 and Table A-2 in the appendix A provide the results for specified test scenario in Chapter 5. Having pass all the test cases which were captured previously, this provides good overview about the GUI functionality.

In addition to the systematic testing approaches the developed prototype was shown to senior people in software industry (including project managers, and senior tech leads) and asked how do they interpret these generated results as experts in the domain. Appendix I includes their answerers. Ten domain experts were selected for this questioner who are more concerning on SPS and who are facing the problems due poor SPS. As all of them have highlighted that they are happy about the quality of the proposed solution by the system (including their remarks and criticism on the system) and identified the further improvements that is required to bring this work into production. Therefore, all these results show that the proposes approach is capable of handling the complexity in SPS.

6.1 Retrospective analysis of project scope and what is delivered

The main aim of this project is to generate a rational project schedule as an intelligent system for software project management. Having this aim, many objectives were identified to fulfil when completing this project in the Section 1.2. Those are namely as follows:

(1) *To provide a literature review of various approaches on project scheduling*
It is required to explore the current state of the research related with project scheduling. This provides to get a good understanding of what are the strengths and weaknesses of each methodology and to get a good intuition to develop a working methodology for project scheduling.

In this research project a substantial literature review was conducted and presented the findings in Chapter 2. This includes main areas of research: 1) SPS with ant colony optimization, 2) Multi-project scheduling with priority rules and analytic hierarchy process, 3) Simulating annealing based project scheduling, 4) Software project scheduling with Bayesian Networks, 5) Multi-agent optimization algorithm for SPS, and 6) Generic algorithm based SPS. For each area detailed literature findings were presented together with in each strength and weaknesses. Furthermore, a condensed summary of these findings were presented in Table 2-1. After this review, it was possible to conclude that GA based approach for SPS is much better and even having past evidences with acceptable quality of results.

(2) *Identifying the most important factors affect on software project scheduling*

There may be many factors that may important in software project scheduling. It is important to identify which are the most important factors in SPS in general. As an example, in Sri Lanka, number of working hours of a software developer is not always 8Hrs per day, therefore, contribution from an employee may more than 100%.

SPS being a complex project, its complexity can be explored through its factors. Therefore, this pint was identified as very important and identified many factors that affect on this. Namely: 1) Employee skills, 2) Employee Salary, 3) Employee dedication, 4) Required skills in a task, 5) Required effort for a task, 6) Precedence graph, 7) Total cost of a solution, 8) Total project duration of a solution, 9) Number of tasks with no employee appointed, 10) Number of skills missing in order to perform all project tasks, and11) Overwork of the project. Full details about each of these factors and their formalisation presented in Section 3.2. Furthermore, having interviews with domain experts (mainly when collecting the questioner to get there feedback on proposed system) also confirmed that these are the most important factors (they highlighted some few more features that can be used as future improvements).

(3) *Developing an approach for intelligent project scheduling*

As the main technology genetic algorithm (GA) will be used, how GA can be used in a project scheduling is the main outcome of this objective.

This is the heart of this research and a detailed explanation on postulated approach presented in Chapter 3. As GA was selected, the problem of SPS needs to be fit into this technology and mainly the approach explains this process. This includes encoding software schedule into a chromosome representation, generating initial population, introducing selection techniques, applying genetic operators with appropriate probabilities, evaluating the fitness of each individual in a population, let the population to evolve as natural evaluation in the mother nature. In each phase the most important mechanisms or technique was selected. For the encoding binary encoding was used, therefore; it was very easy to apply genetic operations (specially mutation) and uses less memory to higher number of computations throughout the evolution process. Population was created through a random process that gave non biased individuals. Furthermore, in this regard, implementation guidelines were provided to select the size of the population (see Section 3.3.1). Having large or smaller population size it was required to change the crossover and mutation probabilities as per the research findings (see Section 3.3.1). As the selection technique tournament selection was used which has the potential to converge the population towards necessary optimisation directions. This selection method has recommended by many researchers and have shown the potential in other GA based approaches too. Furthermore, elicit selection is also used in this process as it always makes sure that the best individual in each population moves to the next population. As GA operations crossover and mutation were used with necessary probability values as suggested by many researchers. Finally, an integrated equation from both cost and penalty for the derived solution used as the fitness function. This function mainly act as a minimization problem as the goal of the evolutionary process is to minimise the both software cost and penalty cost.

(4) *Implementing a working prototype*

The proposed approach will be implemented as a prototype by using a suitable programming language (for e.g. in Java). This prototype should use set of constraints specific to a software project as inputs together with other required input data (for e.g. some parameter values). Having this input data it should generate a Gantt chat which is the standard format in software industry for a project schedule.

A successful prototype was developed including a GUI to present the proposed approach. Chapter 4 provides the details for this. As though at the beginning Java became the best programming language and implemented the full system in Java. In the implementation phase many algorithms and data structures were used to achieve the required functionary. Furthermore, many java libraries used to include external, namely: xStream, JFreeChart, and java Swing libraries. Furthermore, as the database SQLite is used.

(5) *Provide a method to input project data into the system and to presents the output data*

Through this it should be possible for a project manager to interact with the system easily.

A GUI also implemented for this prototype, therefore users can easily interact with the system. In additionally XML is used to pass the data into the proposed GA based approach. Due to introducing this XML form to enter the data, that will be very useful for larger projects where it is harder to manually enter each data one by one through the given GUI. User can even edit the XML file directly to provide necessary input data. Finally a Gantt chart also created that shows the final best answer that system derived.

(6) *Evaluate the quality of the system*

System will be tested with actual data of a real software project. This provides the confidence how good the proposed system is and considered as a formal project evaluation.

System was evaluated through a specific generic scenario selected. This scenario is simple enough to show the workings of this approach. Also Turing Test approach is used to evaluate the developed approach and due to this methodology it was easy to identify how this approach is different from real human expert. From the results it was found that system also provides interesting results though it is required further fine tunings to make answers more accurate and effective.

6.2 Future work

The main issue identified in this approach is solutions are not fully optimised. One reason for this is still the fitness equations are not strong enough to derive a better result. If we can improve the scope and features of the fitness equation, it is further possible to optimise the problem. Nevertheless, it is not easy to quickly find such equation(s), but further research will contribute to help on this direction.

In addition to the limitation of fitness function, the current approach not always selecting most practical set of candidates for each task. The reason for this is, in the current approach it is assumes that employees with specific skill always have the same knowledge/expertise. Nevertheless, this is not the reality. Therefore, together with the skills of each employee it is important to consider his/her experience as well (skill level). This will add another factor into the optimisation problem.

Another limitation is for each task it is required to find employees that have the required skills as a team to complete. Nevertheless, it is a question whether all such employees should be allocated from start to end of that task. Sometimes this may not require most of the time. Therefore, this also effects on performance of the approach. This can be eliminated using a more detailed task breakdown or assigning people for multiple task that can execute in parallel.

Overwork information also can be improved, so that it is important to provide specifically which employee was assigned for overwork and to include other characteristics of that employee (e.g., skill level, max dedication, and holiday data) to get a good insight why that employee needs extra work to complete a task. This need more work from GUI perspective.

Another improvement for this work is to introducing dynamic values for mutation and crossover probabilities. In the current approach, always it is used a static value throughout the process. Nevertheless, latest research has found that it is important to use higher rate of mutations at the beginning while making it lower and lower through the time, while to increase the crossover probability over time. When the mutation rate is high it allows to jump from local minima while higher cross over helps to converge in a given minima.

Another good improvement is to increase the usability of GUI. Though this is not key focus of this work, it is very important to have a simple but user friendly interface to use this as an application.

References

[1] B. W. Boehm, ‘Software risk management: principles and practices’, *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991.

[2] R. N. Charette, *Software engineering risk analysis and management*.New York: Intertext Publications : McGraw-Hill Book Co, 1989.

[3] W. Herroelen, ‘Project Scheduling-Theory and Practice’, *Prod. Oper.Manag.*, vol. 14, no. 4, pp. 413–432, Jan. 2009.

[4] K. Ewusi-Mensah, *Software development failures: anatomy of abandoned projects*. Cambridge, Mass: MIT Press, 2003.

[5] J. O. Kephart and D. M. Chess, ‘The vision of autonomic computing’, *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[6] S. Sahay, B. Nicholson, and S. Krishna, *Global IT outsourcing: software development across borders*. Cambridge, UK ; New York: Cambridge University Press, 2003.

[7] A.-M.Søderberg, S. Krishna, and P. Bjørn, ‘Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships’, *J. Int. Manag.*, vol. 19, no. 4, pp. 347–361, Dec. 2013.

[8] C. K. Chang and M. Christensen, ‘A net practice for software project management’, *IEEE Softw.*, vol. 16, no. 6, pp. 80–89, Dec. 1999.

[9] C. Chao, ‘Software Project Management Net: A New Methodology on Software Management’, University of Illinois at Chicago, Chicago, IL, USA, 1995.

[10] K. Schwaber and M. Beedle, *Agile software development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002.

[11] E. Alba and J. Franciscochicano, ‘Software project management with GAs’, *Inf. Sci.*, vol. 177, no. 11, pp. 2380–2401, Jun. 2007.

[12] C. K. Chang, M. J. Christensen, and T. Zhang, ‘Genetic Algorithms for Project Management’, *Ann. Softw.Eng.*, vol. 11, no. 1, pp. 107–139, 2001.

[13] W. Huang, L. Ding, B. Wen, and B. Cao, ‘Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times’, in *Advances in Neural Networks – ISNN 2009*, vol. 5552, W. Yu, H. He, and N. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 60–69.

[14] W.-N.Chen and J. Zhang, ‘Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler’, *IEEE Trans. Softw. Eng.*, vol. 39, no. 1, pp. 1–17, Jan. 2013.

[15] C. N. Bodea, I. R. Badea, and A. Purnus, ‘Complex project scheduling using multi-agent methods: A case study for research projects’, *Manag. Mark.*, vol. 5, no. 3, p. 21, 2010.

[16] W. Chen, Y. Shi, H. Teng, X. Lan, and L. Hu, ‘An efficient hybrid algorithm for resource-constrained project scheduling’, *Inf. Sci.*, vol. 180, no. 6, pp. 1031–1039, Mar. 2010.

[17] J. Xiao, X.-T.Ao, and Y. Tang, ‘Solving software project scheduling problems with ant colony optimization’, *Comput.Oper. Res.*, vol. 40, no. 1, pp. 33–46, Jan. 2013.

[18] M. Dorigo, V. Maniezzo, and A. Colorni, ‘Ant system: optimization by a colony of cooperating agents’, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[19] M. Dorigo and G. Di Caro, ‘Ant colony optimization: a new meta-heuristic’, 1999, pp. 1470–1477.

[20] M. Dorigo and L. M. Gambardella, ‘Ant colony system: a cooperative learning approach to the traveling salesman problem’, *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[21] K. M. Sim and W. H. Sun, ‘Ant colony optimization for routing and load-balancing: survey and new directions’, *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 33, no. 5, pp. 560–572, Sep. 2003.

[22] M. Mitchell, *An introduction to genetic algorithms*. Cambridge, Mass: MIT Press, 1996.

[23] A. Singh, ‘Resource Constrained Multi-project Scheduling with Priority Rules & Analytic Hierarchy Process’, *Procedia Eng.*, vol. 69, pp. 725–734, 2014.

[24] R. W. Saaty, ‘The analytic hierarchy process—what it is and how it is used’, *Math. Model.*, vol. 9, no. 3–5, pp. 161–176, 1987.

[25] K. Bouleimen and H. Lecocq, ‘A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version’, *Eur. J. Oper. Res.*, vol. 149, no. 2, pp. 268–281, Sep. 2003.

[26] E. H. L. Aarts and J. Korst, *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. Chichester [England] ; New York: Wiley, 1989.

[27] V. M. Dalfard and V. Ranjbar, ‘Multi-projects scheduling with resource constraints and priority rules by the use of simulated annealing algorithm’, *Tech. Gaz.*, vol. 19, no. 3, pp. 493–499, 2012.

[28] V. Khodakarami, N. E. Fenton, and M. Neil, ‘Project scheduling: improved approach to incorporate uncertainty using Bayesian networks’, *Proj. Manag.J.*, vol. 38, no. 2, pp. 39–49, 2007.

[29] X. Zheng and L. Wang, ‘A multi-agent optimization algorithm for resource constrained project scheduling problem’, *Expert Syst. Appl.*, vol. 42, no. 15–16, pp. 6039–6049, Sep. 2015.

[30] C. K. Chang, H. Jiang, Y. Di, D. Zhu, and Y. Ge, ‘Time-line based model for software project scheduling with genetic algorithms’, *Inf. Softw. Technol.*, vol. 50, no. 11, pp. 1142–1154, Oct. 2008.

[31] S. J. Russell, P. Norvig, and E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.

[32] S. Hartmann and D. Briskorn, ‘A survey of variants and extensions of the resource-constrained project scheduling problem’, *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 1–14, Nov. 2010.

[33] L. Ozdamar, ‘A genetic algorithm approach to a general category project scheduling problem’, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 29, no. 1, pp. 44–59, Feb. 1999.

[34] V. Yannibelli and A. Amandi, ‘A knowledge-based evolutionary assistant to software development project scheduling’, *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8403–8413, Jul. 2011.

[35] K. A. De Jong, ‘An Analysis of the Behavior of a Class of Genetic Adaptive Systems.’, University of Michigan, Ann Arbor, MI, USA, 1975.

[36] J. Grefenstette, ‘Optimization of Control Parameters for Genetic Algorithms’, *IEEE Trans. Syst. Man Cybern.*, vol. 16, no. 1, pp. 122–128, Jan. 1986.

[37] J. D. Schaffer, R. Caruana, L. J. Eshelman, and R. Das, ‘A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization’, in *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1989, pp. 51–60.

[38] L. Davis, Ed., *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.

Appendix A

Test cases and expected results on related functionalities on employees can be found in below table.

Table A-1: Employee information related test cases

Test Case ID	Test Description	Test Steps	Test Data	Expected Results	Status
TC_ED_01	Add employee details	1.Fill the Employee ID 2. Fill the Employee Name 3. Fill the Designation 4. Fill the Salary 5. Fill the skill 6. Fill the dedication 7. click on Add button	Employee ID: 01 Employee Name: TharuWijerathne Designation: QA Engineer Salary: 110000 Skill: Manual test, Test Automation Dediation:1.2	Added employee details should display on the employee details table	Pass
TC_ED_02	Edit employee details	1. Select a employee 2. change the dedication of the employee	Employee Name: TharuWijerathne Designation: QA Engineer Salary: 100000 Skill: Manual test, Test Automation Dedication: change 1.2 to 1.0	Dedication should be appear as 1.0	Pass
TC_ED_03	Delete employee details	1. Select a employee 2. click on delete button		Selected employee should disappear on the table	Pass
TC_ED_04	Add skill	1.Fill the skill ID 2. Fill the skill Name 3. click on Add skill button	Skill id: 01 Skill Name : QA	Added skills should display on skill table.	Pass

Functionalities of the task details GUI includes 1) Add task, 2) Edit task, and 3) Delete task. To cover these functionality of the test plan is presented in below table.

Table A-2: Task information related test cases

Test Case ID	Test Description	Test Steps	Test Data	Expected Results	Status
TC_ED_01	Add Task details	1.Fill the Task ID 2. Fill the Task Name 3. Fill the Task Effort 4. Fill the required skill 5. Select the Dependency	Task ID: T5 Task Name: QA Task Effort: 25 Required skill: Test Automation Task Dependency: T4	Added task details should display on the task details table	Pass
TC_ED_02	Modify Task Details	1. Select a task 2. Change the effort to task	Task ID: T5 Task Name: QA Task Effort: change effort 25 to 35 Required skill: Test	Effort should appear as 35	Pass

			Automation Task Dependency: T4		
TC_ED_03	Delete Task details	1. Select a task 2. click on delete button		Selected task should disappear on the datable	Pass

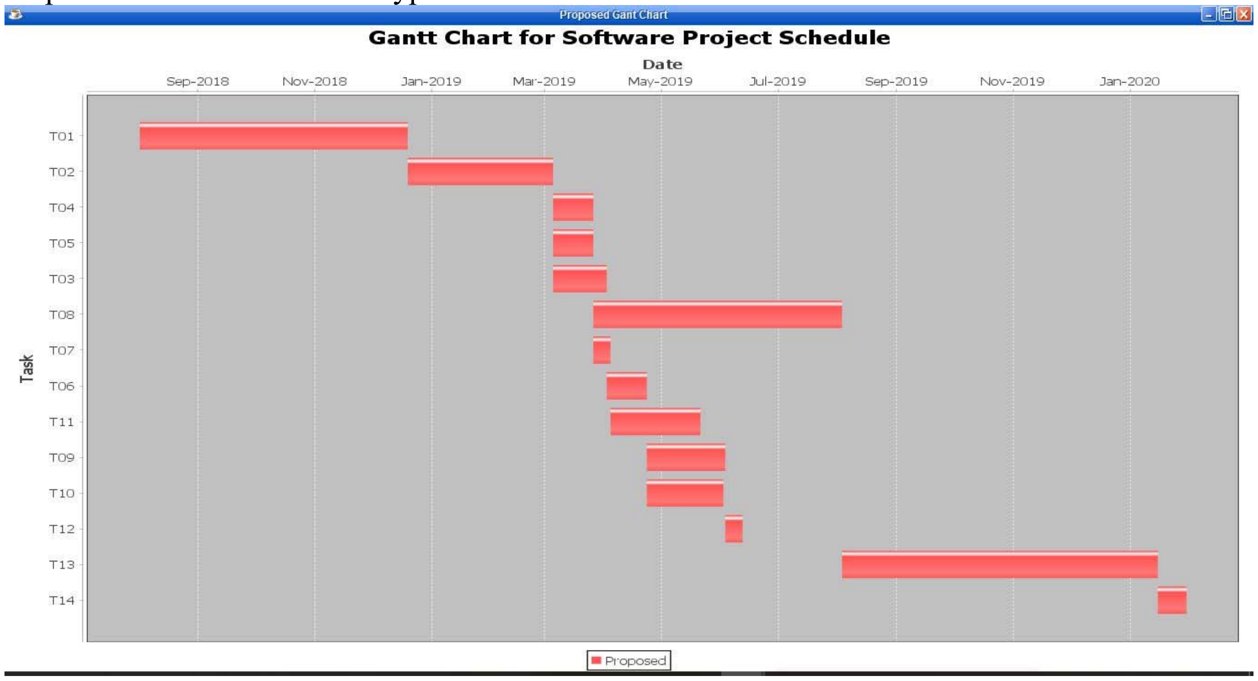
Appendix B

This appendix includes the questionnaire developed for the Turing Test approach.

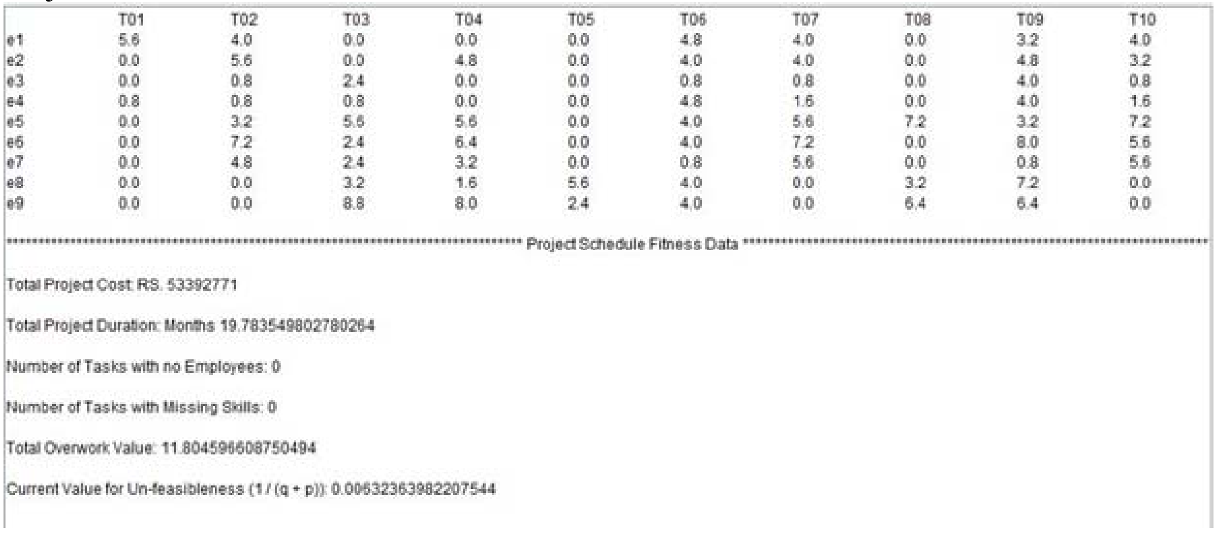
Prototype Evaluation form for Rational Intelligent Project Scheduler for Software Project Management

For this questioners I have generated a sample project plan and attached the Gantt chart for your reference. The same constrains I have used to prepare this chart is given in below as task table and employee details table. Please fill the following questioners after preparing your own project schedule.

Proposed Solution from Prototype



Project Schedule Fitness Data



```
        T01     T02     T03     T04     T05     T06     T07     T08     T09     T10
e1      5.6     4.0     0.0     0.0     0.0     4.8     4.0     0.0     3.2     4.0
e2      0.0     5.6     0.0     4.8     0.0     4.0     4.0     0.0     4.8     3.2
e3      0.0     0.8     2.4     0.0     0.0     0.8     0.8     0.0     4.0     0.8
e4      0.8     0.8     0.8     0.0     0.0     4.8     1.6     0.0     4.0     1.6
e5      0.0     3.2     5.6     5.6     0.0     4.0     5.6     7.2     3.2     7.2
e6      0.0     7.2     2.4     6.4     0.0     4.0     7.2     0.0     8.0     5.6
e7      0.0     4.8     2.4     3.2     0.0     0.8     5.6     0.0     0.8     5.6
e8      0.0     0.0     3.2     1.6     5.6     4.0     0.0     3.2     7.2     0.0
e9      0.0     0.0     8.8     8.0     2.4     4.0     0.0     6.4     6.4     0.0

************************************** Project Schedule Fitness Data **************************************

Total Project Cost: RS. 53392771

Total Project Duration: Months 19.783549802780264

Number of Tasks with no Employees: 0

Number of Tasks with Missing Skills: 0

Total Overwork Value: 11.804596608750494

Current Value for Un-feasibleness (1 / (q + p)): 0.006323639822207544
```

Task data and Task Dependencies

Task #	Task	Task effort	Skill Required	Task Dependencies
T1	Business analysis	30	BA,	
T2	System design	65	IT Application architecture, Java,	T1
T3	DB Implementation	30	Oracle, SQL	T2
T4	UI Design	15	UI/UX, SQL, Node.js,	T2
T5	Create Test Case	10	Manual QA,	T2
T6	Core Implantation	25	Java, Oracle, SQL	T3
T7	User Interface Review	10	UI/UX, BA, Java	T4
T8	Create Automation Framework	45	Selenium, SQL, J meter, TestNG	T5
T9	Implement Feature 1	45	Java, SQL, Oracle, Data structures	T6
T10	Implement Feature 2	35	Oracle, Java,	T6
T11	UI Implantation	45	UI/UX, SQL, Java, Node.js	T7
T12	System Integration	10	Java, SQL, Oracle, Node.js	T9, T10, T11
T13	Integration Testing	75	Selenium, SQL, TestNG, Manual QA, SQL, J meter	T12, T8
T14	UAT	5	BA	T13

Employee Data

ID	Employee name	Designation	Salary (LKR)	Skills	Dedication
e1	Nimal Fernando	Business Analysis	100000	BA, Java,	6
e2	Mahesh Perera	Senior Software Engineer UI	150000	Java, UI/UX, Spring	8
e3	Ranil Jayamaha	Senior Architect	450000	Java, Oracle, Spring, Modeling skills, Project management skills, Marketing skills, Critical reasoning skills, IT Application architecture	4
e4	Indiaka Samarasinghe	Teach Lead	250000	Java, Oracle, Spring, Design patterns, Code reviews, Version control, Team planning	8
e5	Kapila Somabandu	Release Engineer	110000	Java, Oracle, SQL, Php, Node.js, Data structures, Debug skills	8
e6	Maven Dabare	Software Engineer	110000	Java, Oracle, Node.js, Data structures	8
e7	Ariyapala Hetiarachi	Software Engineer	100000	Java, Oracle, Php, Node.js	6
e8	Tharu Wijerathne	Software Quality Assurance Engineer	110000	Selenium, SQL, J meter, TestNG, Manual QA	8
e9	Lakmali Thisera	Quality Assurance Engineer	85000	Selenium, SQL, Manual QA	10

Q1) How long did you take to finish your schedule

- 30 mints
- 45 mints
- More than one hour

Q2) How easiest to generate a new different schedule for the same data set.

- Easy
- Moderate
- Hard

Q3) How better the project duration in what you have drawn manually relative to the given reference project schedule. Manually drawn schedule duration = t1 Reference schedule duration = t2

- t1 > t2
- t1 < t2
- t1 = t2

Q4) Total project cost also mainly assumed as the salaries of employees. So how Better the project cost in what you have drawn manually relative to the given reference project schedule. Manually drawn schedule cost = c1 Reference schedule cost = c2

- c1 > c2
- c1 < c2
- c1 = c2

Q5) Calculate the total over work for the schedule and compare with the reference given Manually calculated over work - 01 Reference over work- O2

- o1 > o2
- o1 < o2
- o1 = o2

Q6) Is there any task in your schedule with no employees assign

- Yes
- No

Q7) Is there any task in your schedule with missing skills

- Yes
- No

Q8) Do you think whether the software project scheduling system can be automated

- Yes
- No

Q9) If it is possible as for your experience how hard it would be to automate this problem

- Easy
- Moderate
- Hard

Q10) If there is a such a system will you be used in your project scheduling work

- Yes
- No

Q11) Comments and/or questions