An Uncertainty Based Genetic Algorithm Approach

for Project Resource Scheduling

جدولة موارد المشاريع باستخدام الخوارزمية الجينية

by

SAIF ALKETBI, MSc

A thesis submitted in fulfilment

of the requirements for the degree of

DOCTOR OF MANAGEMENT

at

The British University in Dubai

Professor Halim

OCTOBER 2016

© Saif Alketbi 2016; British University in Dubai; UAE

Reproduced with permission of the copyright owner. All rights reserved.

An Uncertainty Based Genetic Algorithm Approach for Project Resource Scheduling

جدولة موارد المشاريع باستخدام الخوارزمية الجينية

by

SAIF ALKETBI

A thesis submitted to the Faculty of Management
in fulfilment of the requirements for the degree of
DOCTOR OF MANAGEMENT
at
The British University in Dubai
October 2016

Thesis Supervisor
Professor Halim

Approved for award:

Name Designation	Name Designation
Name Designation	Name Designation

Date: ____________

DECLARATION

I warrant that the content of this thesis is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that one copy of my dissertation will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make that copy available in digital format if appropriate.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

____

Signature

COPYRIGHT AND INFORMATION TO USERS

The author whose copyright is declared on the title page of the work has granted to the British University in Dubai the right to lend the thesis to users of its library and to make partial or single copies for educational and research use.

The author has also granted permission to the University to keep or make a digital copy for similar use and for the purpose of preservation of the work digitally.

Multiple copying of this work for scholarly purposes may be granted by either the author, the Registrar or the Dean of Education only.

Copying for financial gain shall only be allowed with the author's express permission.

Any use of this work in whole or in part shall respect the moral rights of the author to be acknowledged and to reflect in good faith and without detriment the meaning of the content, and the original authorship.

ABSTRACT IN ENGLISH

This research is tackling the issue of complex resources scheduling in project management. In traditional planning tools, resource allocation is sequence based. This normally results in a very simple baseline schedule. However, in reality, the problem of project scheduling is more complex and it depends on a multitude of factors. For example, project scheduling when combined with resources constraints and activities duration uncertainty is an interesting research problem that has recently has attracted the effort many researchers. Previous research has developed a simulation-based approach to solve the problem by optimizing resources resource allocation decisions on starting specific project activities at specific times. Several nonlinear optimization models were developed for this purpose assuming uniform resource availability and sequence based project tasks. The work presented in thesis add to the existing literature in a proposing the use of a genetic algorithm uncertain approach to resource-scheduling in projects. This research focuses on one of the most important aspects, which is uncertainty. The uncertainty aspect was not incorporated effectively in in the previous resource modeling models. The uncertainty of time estimation is one of the most important problems which reduce any resource scheduler effectiveness. Genetic algorithm was chosen as the main methodology to build resource scheduler. The results showed the proposed methodology outperformed existing algorithms in optimizing project durations and resources allocation. The main contribution of the proposed scheduler is its ability to incorporate uncertainty in scheduling process. Results proofed effectiveness and outperformance of the proposed solution. The genetic algorithm was tested on several projects from the existing databases and on one new project to the validity of the approach. The proposed algorithm out

performed fairly well the results that exists from previous studies. One major contribution of this research is the incorporation of uncertainty to optimize project duration based on resource allocation.

ABSTRACT IN ARABIC

هذا البحث يتصدى لجدولة الموارد المعقدة في إدارة المشاريع. بالنسبة لأدوات التخطيط التقليدية، تخصيص الموارد مبني على أساس التسلسل. هذا ينتج عادة جدولة بسيطة جدا. ومع ذلك، في الواقع، مشكلة جدولة المشاريع أكثر تعقيدا و هي تعتمد على العديد من العوامل. على سبيل المثال، جدولة المشاريع عندما تمتذج مع عدم اليقين تصبح مشكلة بحث مثيرة للاهتمام بحيث تستقطبت في الآونة الأخيرة جهود العديد من الباحثين. وقد وضعت الأبحاث السابقة النهج القائم على المحاكاة لحل المشكلة عن طريق الاستفادة المثلى لقرارات تخصيص الموارد على بدء أنشطة المشروع وفق أوقات محددة. وقد وضعت عدة نماذج للتحسين غير الخطية لهذا الغرض على افتراض توافر الموارد بشكل موحد و اقرار مهام المشروع بشكل متسلسلا. العمل المقدم في هذه الأطروحة استخدم نهج الخوارزمية الجينية في جدولة المشاريع. يركز هذا البحث على واحد من أهم الجوانب، وهو عدم اليقين. لم يتم دمج جانب عدم اليقين بشكل فعال في نماذج جدولة الموارد السابقة. عدم التيقن من تقدير الوقت هي واحدة من أهم المشاكل التي تقلل من فعالية أي جدولة للموارد. وقد تم اختيار الخوارزمية الجينية كمنهجية رئيسية لبناء جدولة الموارد. أظهرت النتائج المنهجية المقترحة تفوق الخوارزميات المقترحة في تحسين فترات المشروع وتخصيص الموارد. المساهمة الرئيسية للجدولة المقترحة هي قدرتها على دمج عدم اليقين في عملية الجدولة. النتائج تثبت الفعالية والأداء المتفوق للحل المقترح. تم اختبار الخوارزمية الجينية على عدة مشاريع من قواعد البيانات الموجودة وعلى مشروع واحد جديد لتاكيد صحة هذا النهج. تمكنت الخوارزمية المقترحة من أداء جيدا مقارنتة مع النتائج الموجودة من الدراسات السابقة. المساهمة الاساسية لهذا البحث هو إدماج عدم اليقين لتحسين مدة المشروع بناء على تخصيص الموارد.

DEDICATION

To My Family

ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to Prof. Halim for his support and guidance throughout my research. His insight and teachings were essential to complete my research. In addition, I am very thankful to Prof. Paul who provided me with enormous help during proposal phase. He pointed me in the right directions to find the most interesting problem statement which is the corn stone of this thesis. Finally, words cannot express my gratitude to my family for their support and patience throughout my PhD study.

Table of Contents

Abbreviations

ALB	assembly line balancing problem
ALV	probability mapping due to Alvarez-Valdes
AMA	alter mode-assignment (scheme)
ANOVA	analysis of variance
AON	activity-on-arc network
AON	activity-on-node network
B&B	branch-and-bound
BAK	probability mapping due to Baker
BFS	best fit strategy
BOM	bill of materials
BP	bin packing problem
BR	backward recursion
BRS	biased random sampling
CIM	computer integrated manufacturing
CMA	change mode-assignment (scheme)
COP	probability mapping due to Cooper
CPM	critical path method
CPU	central processing unit
DA	disjunctive arc procedure
FFS	first fit strategy
FMA	first mode-assignment (scheme)
FMS	flexible manufacturing system
FR	forward recursion
FSP	flow shop problem
GA	genetic algorithm
GERT	graphical evaluation and review technique
GPB	general precedence-based bound
GRB	general resource-based bound
GRD	greatest resource demand (priority rule)
GRPW	greatest rank positional weight (priority rule)
GRPW	greatest rank positional weight (priority rule)
IMA	improved mode-assignment (scheme)
IRSM	improved RSM (priority rule)
JSP	job shop problem
LCBA	local constraint based analysis (priority rule)
LFT	latest finish time (priority rule)
LFT	latest finish time (priority rule)
LS	local search procedure
LST	latest start time (priority rule)
LST	latest start time (priority rule)
MAP	mode assignment problem
MIP	mixed integer program

MIRCPSP	mode identity resource-constrained project scheduling problem
MIS	most immediate successors (priority rule)
MMH	multi-mode heuristic
MMPSP	multi-mode project scheduling problem
MP8	master production schedule
MPP	multi-pass procedure
MRBRS	modified regret based biased random sampling
MRCPSP	multi-mode resource-constrained project scheduling problem
MRP	material requirements planning
MSC	modified slack calculation
MSLK	minimum slack (priority rule)
MSLK	minimum slack (priority rule)
MTS	most total successors (priority mle)
MTS	most total successors (priority rule)
NBRS	normalized biased random sampling
NPV	net present value
OMA	opening mode-assignment (scheme)
PERT	program evaluation and review technique
PFSP	permutation flow shop problem
PPB	parallel precedence-based bound
PPC	production planning and control
PRB	parallel resource-based bound
PSP	project scheduling problem
PSPDL	project scheduling problem with given
PSPST	project scheduling problem with setup times
PSS	parallel scheduling scheme
RCPSP/T	resource-constrained project scheduling problem with time
RCPSP	resource-constrained project scheduling problem
RS	random sampling
RSM	resource scheduling method
SA	simulated annealing
SGS	schedule generation scheme
SMDLH	single-mode deadline heuristic
SMH	single-mode heuristic
SMPSP	single-mode project scheduling problem
SPB	serial precedence-based bound
SPP	single-pass procedure
SPT	shortest processing time (priority rule)
SSS	serial scheduling scheme
STH	setup time heuristic
TBB	truncated branch and bound procedure
TS	tabu search

WCS	worst case slack (priority rule)
WCS	worst case slack (priority rule)

List of Figures

List of Tables

Chapter 1: Introduction to the Research

1.1 Introduction

In this chapter introduces the key research theoretical background concepts and the scientific drivers behind this research. The chapter presents the rationale behind the research problem. The chapter further, introduces the research questions and objectives that guided this research to completion. Also, the contribution of this research presented. The last section of the chapter presenters a general overview of the thesis structure

Resource scheduling issue has been studied intensively recently. Several methods have been proposed by authors to approach this issue. Each one of these authors approached the problem from different angles. Gomez et al (2014) asked how the objective of resource scheduling process may affect the performance of projects in their portfolio. They cited number of objectives usually used by managers in the numerous industries. Resources scheduling based on each one of these objectives will lead to different overall projects performance. For example, if the objective of resources scheduling is to shorten lifespan of all projects, this may lead to reducing resource utilization throughout these projects progress. Also, if the main resources scheduling objective was reducing costs, this may lead to a longer lifespan of all projects. The same question has been asked by Wang et al (2014). However, the authors in the later work introduced the uncertainty of effect in resource scheduling process. On the other hand, Dutra et al (2014) were wondering what is the best probabilistic model to prioritize projects in multi-projects environment. Similarly, Messelis et al (2014) wondered about prioritizing different resources scheduling algorithms.

1.2 Problem Statement

All of the mentioned references used a combination of heuristic and meta-heuristic approaches to tackle resources scheduling issue. They were able to produce good algorithms and techniques to address resources scheduling issue from their proposed perspective. However, their assumptions may be unrealistic in general situations. For example, Gomez et al (2014) and Wang et al (2014) works assumed that the selection of resources scheduling objectives is decided by the manager preferences only. This wide assumption is unrealistic because the available amount of resources may have a direct relationship with the selection of resources scheduling objective. If resources are very limited, choosing an objective that has more emphasis on reducing required resources is more realistic. Similarly, Dutra et al (2014) developed probabilistic model to prioritize projects. Assuming that all variables in prioritizing projects are based on stochastic process is very unrealistic.

1.3 Research Questions

On the basis of the above background and the problem statement, this research attempt to answer the following question:

- To what extent the uncertainty problem affect scheduling process in project management?
- Can meta-heuristic approach such as genetic algorithm provide a better solution to the problem resource scheduler under uncertainty?
- What is the best approach to utilize meta-heuristic optimization techniques in performing scheduling process in project management?

1.4 Research Objective

The main aim of this thesis is to address the issues raised in literature such as uncertainty in wider project management situations. The following are the main objective of this research:

- To tackle resource scheduling in project managements in a way that addresses shortcomings in literature especially uncertainty effect.
- To use advanced meta-heuristic approach such as genetic algorithm to build an automated resource scheduler under uncertainty.
- To evaluate and test proposed approach based on adopted standards in literature so that valid comparison can be established.
- To evaluate the proposed approach based on real resource scheduling task in real project management situation.

The proposed approaches would be enhanced and improved so that they can handle harder problems. Numerous ways of combining these approaches is proposed. Also, artificial intelligence will be utilized to have smarter mechanisms of resources scheduling in project portfolio management. These new proposed mechanism should be able to outperform the performance of the solutions proposed in the literature.

1.5 Contribution to the body of knowledge

This thesis provides contribution to knowledge in the following areas:

- Understanding the complexity of resources scheduling under uncertainty.
- Resources scheduling approach for project management which based on Genetic Algorithm. An encoding technique to utilize binary Genetic Algorithm without extensive modifications as seen in literature.

- A simulation technique to simulate scheduling process computation in a way that allows utilization of meta-heuristic approaches was implemented.

1.6 Research Flow layout

This research is divided into phases. Each one of these phases composes of a set of activities. The order in which these activities are performed depends on the nature of the activity and the progress achieved so far in the research. The following subsection describes these phases in more details and highlights the general objectives of each research activity.

1.7 Research Design and Methodology

As said before, this research will use qualitative and quantitative research approaches to develop good resources scheduling mechanisms for project portfolio management. Both of the analysis and development require a lot of work in term of gathering the needed knowledge.

To summarize this research design and methodology, it can be discussed as multi stage project. In the first phase, comprehensive literature review will be performed. Then, necessary background on mathematical tools and artificial intelligence techniques will be acquired. Later on, proposed theory is formed which will be used to model and implement the proposed solution. Lastly, extensive analysis and discussion was performed to lay down this research contributions and conclusions.

1.7.1Literature Review

This activity is one of the most important activities in this research. Despite that it starts in the first phase, it will keep being conducted throughout the research. Here, all the related works in literature should be reviewed, summarized and criticized. Some works may have a good solutions and insights which can be used in other stages in this research. Also, authors of interesting works can be contacted to start discussion regarding their works. Important questions can be easily answered by these authors and beneficial cooperation can be established.

This activity requires an access to well-known scientific databases such as ScienceDirect. If the university library does not have the required subscriptions to these databases, an individual subscription may be purchased. These scientific databases allow researchers to look for the related works and access their full text. Another way to look for the related works in literature is to use the free indexing services such as Google Scholar. However, these services do not provide full text of the work. All of these tools will be essential to conduct literature review activity.

1.7.2Mathematical Background

The qualitative analysis in this research requires deep mathematics knowledge of several mathematical theories. The researcher should be able to fully understand the basic elements related to mathematical programming techniques. These techniques are fully essential to optimize resources scheduling in project portfolio management. Note that the researcher may not need to provide any new theorem or to prove any existing mathematical hypothesis in this activity. However, the researcher should fully understand the detailed aspects of how each used theorem of mathematical programming techniques operates.

This activity requires reading many well-known textbooks in mathematical programming field. These textbooks are postgraduate material. Therefore, they may not be found in regular university textbook stores. Mostly, they would be ordered through the Internet. Also, there is a lot of material regarding mathematical programming usage in other fields such as engineering. Online courses covering these topics are abundant. It may be useful to enrol in some of these online courses to acquire the required familiarity with the topic.

1.7.3Artificial Intelligence Background

Similar to mathematical programming background, a good background in artificial intelligence is required so that the researcher can be able to develop and employ the quantitative research approach in this work. This activity should be started after most work in the previous activity has concluded because artificial intelligence techniques analysis depends hugely on mathematical programming. As said before two important techniques of artificial intelligence would be would be used in this research. These techniques are artificial neural networks and genetic algorithms. Mastering these techniques is not expected to be very hard.

1.7.4Proposed Theory Formation

This is one of the most important activities in the whole research. After acquiring the necessary knowledge regarding mathematical programming and artificial intelligence techniques, the researcher will be able to employ this knowledge to develop theoretical basis for the proposed resources scheduling mechanisms. Hypotheses on how resources scheduling mechanisms should operate would be proposed during this activity. The general algorithmic behaviour of the proposed resources scheduling mechanisms will be developed during this activity. These algorithmic behaviours can be presented as pseudo code or sequential executing steps of instructions. In addition, guidelines of how the proposed mechanism can be implemented in reality will be identified as a result of this activity.

1.7.5 Modeling and Simulation

The proposed mechanism is modelled using simulation approach. The reason behind using this approach of testing is to investigate the abstract performance of the proposed mechanisms without the effects of other factors which are found in reality such as manager preferences. Sometimes good resources scheduling mechanisms can be developed which have very good performance on paper. However, managers in the real world do not adopt these mechanisms because they do not understand them or they cannot incorporate them in their management style. This issue (adoption in real work place) will be addressed in the next phase. Nevertheless, the performance of the proposed resources scheduling mechanisms has to be proven before they can be introduced in real project portfolio management environment. This is the sole purpose of this phase.

1.7.6 Results Generation

An actual implementation of the proposed resources scheduling mechanisms using Matlab environment is performed. Every aspect of the proposed resources scheduling mechanisms can be traced and evaluated in very efficient way. A lot of these traced data will be converted into information that describes the evacuated system. Further investigation on the generated information led to forming this research results.

1.8 Thesis Organization

This thesis is organized as follow. It starts with deep literature review on resource scheduling in project management in chapter 2. Then, an extensive discussion about existing scheduling techniques is presented in chapter 3. The discussion is extended in chapter 4 to establish scheduling scope within project management. Research methodology is presented in chapter 5. Afterwards, the proposed solution is evaluated in chapter 6. Finally, the thesis concludes with chapter 7 where conclusions of research are provided.

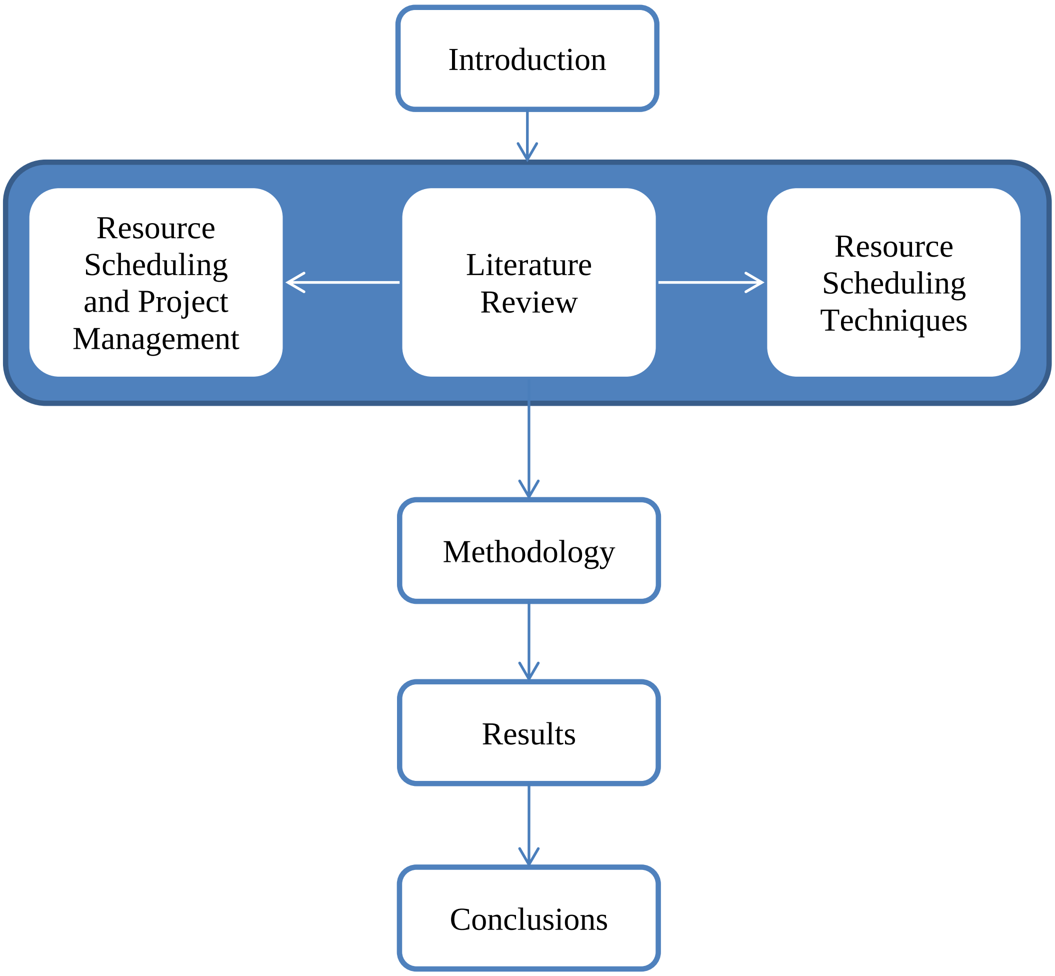


Figure 1.1: Thesis layout.

Chapter 2: Literature Review of Project Scheduling

2.1 Introduction

In modern world projects are present everywhere (Turner, 2014). Projects are present in all kinds of industries due to their flexibility and uniqueness whether it's the sphere of business or personal life (Kerzner, 2013). Projects can also be seen as forming a major part of public and private organizations. In the modern economic world, where products have very small life cycle and the tastes of the consumers rapidly change most of the organizations derive a large share of their revenue through projects. Some businesses have taken this phenomenon a step farther and their entire revenue stream is driven by projects. Where the traditional business tends to repeat, the revenue generating activities in hope that they would continue to add value to the organization, but projects tend to be temporary and have their own unique characteristics (Meredith and Mantel, 2011).

Projects tend to have a lot of common features; some projects require immense resources and investment while others are done within weeks or days without any major investment. In order to understand the true nature of projects, we outline the two following definitions that are widely accepted in the modern economic world (Heagney, 2012).

- Project is defined as a series of actions or activities carried out with the intention of fulfilling or achieving a certain goal or objective. Projects have a discernible start and end and tend to have limited amount of resources in human and non-human form. They are flexible and can serve multiple functions in the organization.

- According to the second school of thought a project is an endeavour of limited time carried out in order to create unique products, services or value adding activities for the organization.

In both definitions a few characteristics of projects become very evident, it is clear that projects are unique activities that are limited in scope and require resources in different forms. Another point to note in the above definition is that projects require cash flows and human resources to complete. The activities and actions of a certain project are driven by the organizational goals or technical capabilities of the organization. Most projects are dynamic and require the activities to be carried out in sync with the time and resources available (Gido and Clements, 2014). Furthermore, most projects tend to have a limited amount of resources available that drives a dependence of activities upon each other that could give rise to certain precedencies between different activities in a project. Due to the temporary nature of projects, they tend to require a specialized team that is responsible for achieving multiple goals for the organization and those achievements can be measures as well.

Project management refers to a coordinated planning and control effort through which all the activities of a project are directed or monitored (Vanhoucke, 2012). Project management deals with the allocation of resources, achievement of objectives, and timing of certain actions and finalization processes that would be required to complete a project successfully. Project management provides a direction for the team and is essential for the successful completion of the project. As per the definition of Project Management Institute (PMI, 2013), project management is recognizing the different requirements of the project, creating clearly identifiable objectives and allocation of resources in a just way that ensures that the maximum interests of all stakeholders are met. A successful project management requires that a project is completed within the estimated time and consumes the resources effectively in accordance with the wishes of the client to achieve different objected that were intended out of the project.

From the perspective of management, a project has five unique stages that involve different management actions in order to successfully achieve its objectives (Gido and Clements, 2014). First stage of a project is called project conception, where there is a lack of concrete information and through the feasibility studies and other means the management decides whether the project should be undertaken or not. At the second stage of the project called Project Definition phase the project's goals, organization and objectives are clearly outlined. In addition to the goals this phase also defines the timeline of the project in form of a roadmap. At the next stage called the Project Planning Phase the project is broken down into smaller activities that would be required to complete the project, then for each activity the resources are allocated in form of Time, manpower and material and an estimation of the cost associated with the execution of these activities. Furthermore, any dependencies for each activity are defined and a finalized schedule is created for the project. These three initial phases concentrate on the planning and resource allocation for the project after which the project finally enters into its execution phase. The management constantly monitors the project in order to determine whether the project is operating as planned or some variances have risen that require corrective actions. Furthermore, the project is also evaluated in term of the quality that helps the management understand the performance level of the project. At the final stage called Termination phase the project is document and finally reaches its conclusion.

Project scheduling mainly deals with the methods of execution for each activity and the time frame for each activity is defined as well (Weglarz, 2012). Projects may be faced with scarcity of resources either in form of time or scarcity of other resources such as manpower and finance. In case of limitation of time the management assumes that the activities cannot be performed at the current financial requirements and the project would need to incur higher costs in order to achieve the desired output. While in the resource constraint the organization does not have the required amount of resources or has a limited amount of resources therefore it is extremely important to take resource constraints into account as well along with the time constraints. Time VS cost and resource leveling are primary example of time constraints in a project. These scenarios clearly identify that time constrained issues can arise in lieu with the resource constraints over the allocation and execution of activities.

There are different types of dependencies in the activities of a project called precedencies. In a common precedence incident an activity cannot be executed without completing the primary activity that would initiate the secondary activity. These generalized precedencies identify the minimum and maximum time lag that would occur between the initiations of different activities. While a more commonly occurring precedence activity is called feeding precedence that requires a certain percentage of previous activity to be completed in order to start the next activity. The difference between generalized and feeding precedence becomes clearly identifiable when the duration of activities is not fixed but is rather flexible or when the activities can be stopped or interrupted during their execution.

The term “resource” in this dissertation refers to the pool of identical amount of resources and resources availability is referred in form of capacity of resource. At the project scheduling stage several different kinds of resources are allocated based on the availability and requirement of the activities (Weglarz, 2012). Renewable resources refer to a resource that does not lose its value or ability to add to value to the organization after the initial application, Manpower and machinery are common examples of renewable resources. While non-renewable resources are consumed or lose their value during the application in the project Raw Material and budgets are common example of non-renewable resources. Renewable and non-renewable resources are usually consolidated in a single category called storage resource. The unit of storage provides a more accurate and easily identifiable measure that can be used to determine the cash balance for the project with regards to the distribution and progress of the payment. Resources such as electricity and virtual memory of the computers and other machine systems can be allocated in a continuous stream are called continuous resources. Partially renewable resources refer to the merging of different time intervals in order to estimate the labor requirement for a project.

A common assumption used in the most projects is that the activities cannot be stopped or interrupted during their execution. However, some circumstances might arise it becomes necessary or more beneficial for the organization to split the activities in different stages. Examples of such activities include the scheduled projects that require a certain downtime and cannot be operated without adhering to the downtimes. These scheduling activities can further be divided into other categories based on the time at which the interruption occurs and the time lag after which the activity can be resumed. In a broader category it is assumed that the activities can be stopped and initiated at any time at the convenience of the project, while some activities may require time based or regular intervals in order to function properly.

An important aspect of problem scheduling deals with the different executions options that are available to the management for a certain project. Single-Model problem assumes that the management does not possess the luxury to select the different execution models or an appropriate execution model has been selected that cannot be changed. While in a multi-model scheduling problem the management can consider several different alternatives to complete an activity this model enables the organization to consider different resource and cost based trade-offs that are a common occurrence in the modern economic environment.

With regards to the scheduling objective the management first needs to differentiate between single-criterion and multi-criterions scenarios. In a multi-criterion problem the management is required to use attainment or goal programming models in order to achieve an acceptable solution. The second function can be distributed between regular and non-regular objectives. Regular objective functions are non-decreasing component driven functions carried out at the initial stages of execution phase of activities. In such a situation the activities cannot be started without compromising the time and resource allocation of other activities. That is the sole reason that problems associated with the regular function can be more easily dealt with as compared to the non-regular functions.

Further attributes associated with project scheduling refer to the level of available information in order to decide and understand the impact of a certain decision. A major assumption used in the scheduling is that the management has all the estimates and information and these information inputs would remain unchangeable at the implementation stage. While in the real would the projects are more dynamic and subject to different changes. Therefore it is critical to account for different uncertainties that might arise during the implementation of schedule. This phenomenon gives rise to another issue in form of interval uncertainty when deciding or planning a project schedule. Fuzzy scheduling problems arise when the data is inaccurate or cannot successfully estimate the outputs by using different variables.

Finally the project scheduling concerns can also be associated with the means through which the information is communicated and dispersed among different team members and the number of primary decision makers involved in the process. Most project scheduling planning assume that the information system is centrally operated with equal access to all participants and the primary responsibility of the decision lies with a single individual or in case of multiple decision makers all the decision maker pursue the same goal and objectives. While reality is a bit different from this assumption because in some dynamic projects decentralization may become essential creating a disparity of information sharing among different decision makers leading to separate objectives and activity execution, this issue can only be resolved through a centralized mechanism of information sharing.

2.2 Schedule Approaches

In case of Resources constrained Project scheduling Problem (RCPSP) it becomes extremely difficult for the management to devise a universally acceptable solution that would be applicable on all instances, it becomes impractical and less cost efficient for the organization to attempt such a feat (Slowinski and Weglarz, 2013). The management can use techniques such as Scatter Graphs, algorithms and meta-heuristic approaches in order to devise an acceptable plan. In past decade or so several large organizations and project driver businesses have employed meta-heuristic techniques to achieve better results in the long run, these techniques have proven to be quite effective in case of RCPSP.

Most commonly faced problem in case of RCPSP is the means through which the life-span can be minimized, there have been several studies carried out in order to achieve these results and in result of such studies there are proven approaches that organizations can use. Other factors included in the problem are that the resources are automatically renewed from one period to another and the activities are not pre-emptive, the resources can consumed only through a single mean, organization has clear estimates or data regarding the availability of the resources, the resources requires also remain static from period to period and the processing time for each activity is measurable and clearly known. Many researchers have shown the effectiveness of the procedure through different means including the PSPLIB benchmark.

2.3 Rule of Thumb

Different approaches such as the Minimum latest finish time (Min LFT) or Long Processing (LPT) have been devises in the RCPSP studies (Kolisch, 2013). The priority rule helps the management devise a list that arranges the activities in a feasible order. The list is driven by the priority rule various priority rules are used to determine these lists and each priority list would yield a different result. Using that priority list the management can generate a schedule. Single pass heuristic methods are considered to be the fastest but least reliable.

To improve the reliability of single pass heuristic methods by devising techniques to reduce the optimal gap have been discussed in the literature (Kolisch, 2013). These include Serial and parallel, using active and pro-active approach and implementing double justification processes. These techniques successfully reduce the optimal gap while keeping the computational requirements low.

A schedule can be generated through Serial Schedule-Generation Scheme (SSGS) or Parallel scheduling methods (P-SGS). In SSGC method each activity is listed basses on the serial order and the activities with earliest time and with least resource requirements. While in P-SGS a clock is used to determine the least time consuming activities in the creation of single unit and the activities with the least time and resource feasibility are given precedence. If the list contains multiple activities in the given time the priority is given to the activity that is performed first in the creation of a unit. Each scheduling method would yield a different method. Both scheduling methods can be used simultaneously and the shortest schedule is given more priority.

RCPSP graph can be viewed as a backward or forward problem depending on the requirements. If we choose the backward problem then the last node is assumed to be the first and there on. The same schedule and same problem would yield different results depending on the method used. The schedule with the shortest life span is considered to be most ideal. Using forward and backward technique in combination with the serial and parallel method can create four different methods and these can be used as the final solutions for the problem.

Once a schedule has been formulated using either of the above mentioned techniques a Gantt chart can be used in order to reduce or eliminate the empty spaces in the schedule. This is usually done by analyzing the Gantt chart from bottom up and filling the activities on the right side in empty spaces. This shifting of activities on the Gantt chart further reduces the time span of the schedule. This process of eliminating optimal gaps and densely packing the graph is called backward justification. After the application of backward justification the forward justification is usually applied in order to further reduce the time span of the schedule. In forward justification the activities are analyzed in the forward order and rearranged to ensure that the activities are packed are more densely on the left side. Application of both processes simultaneously is called double or forward and backward justification. This technique can be applied to improve the overall standing of a solution and provide more accurate results. If the optimal gaps are still not perfect then the heuristic techniques can be used to improve the results but that usually occurs in the addition of computational requirements.

2.4 Meta-Heuristic Approach

In past decade or so meta-heuristic approaches have gained immense popularity due to their ability to reduce the optimal gap and create better solutions (Babu and Shah, 2013). The techniques include Genetic algorithms, simulated annealing, tabu search etc. These techniques are considered ideal to solve RCPSP. Single-pass heuristic techniques use determined methods to search for a solution but meta-heuristics comes up with different solutions in a random manner within the space of schedule in order to search for the optimal solution. Different heuristics methods employ a wide range of search techniques in order to determine the perfect solution but the solutions tend to grow increasingly better as the search progresses. But at some point the search needs to be interrupted in order to save the computational time. The stopping criteria can either be determined through CPU time or on the number of solutions that have been analyzed by the system in order to discover the best solution.

Many meta-heuristic approaches have been developed by observing different phenomenon in the nature. For instance genetic algorithm is based on the evolution of species. Similarly, simulated annealing approach is developed by studying the annealing of metals with each other. Ant colony optimization was developed as a direct result of studying the hunting and foraging methods of ants. Before elaborating on the heuristic approached identified above, it is important to make the distinction between Local and global search sometimes also known as Diversification and Intensification.

Each meta-heuristic search is faced with the same type of dilemma regarding how to converge the best solution in the available search space (Local Search). The second issue is how to direct the search engine towards the best neighborhoods amongst all the options and neighborhoods available to the system (Global Search). A technique that can tackle the both issues simultaneously is guaranteed to provide the best results in shortest span of time. But these both conditions are difficult to achieve in a single search engine, because most of the time the algorithms that direct the search towards local search tend to be in conflict with the global search protocols. Meta-heuristic methods and techniques therefore are required to create an acceptable balance amongst both conditions. Therefore the strategies that are devised to fulfill the global search requirements are called diversification while those that concentrate on local search are called intensification.

Almost all meta-heuristic methods must figure out a balance between the local and global search. Most of the times, a method is more relevant for global search while other a different model can be more ideal for local search. To counter that issue a hybrid of a system can be used to figure out an optimal solution such as the combination of simulated annealing and Genetic algorithms. Simulated annealing is better at local search while genetic algorithm excels at global search. One of the biggest challenges faced in the global search is to avoid being stuck in a non-optimal neighborhood. It is important to incorporate strategies in the search engine to discover that it’s stuck in non-optimal area and devise means to escape that non-optimal area. The former discussion clearly identifies that all meta-heuristic systems should have a combination of local and global search along with the instructions to escape a non-optimal neighborhood.

2.5 Genetic Algorithm

Genetic Algorithm is considered to be the most ideal approach for dealing with RCPSP and optimization problem (Dasgupta and Michalewicz, 2013). GA is primarily designed to replicate the methods used by the nature in evolution of the species. In nature each coming generation attempts to improve itself from the previous one by using the survival of fittest technique. GA method is also called population based method. As the search progresses under the GA method the system creates a group of population whereby they are improved with each successive search creating improved generations until the search is stopped or optimal solution is achieved.

When the GA is applied in case of a RCPSP issue the system considers the activities as a set of population, the system then attempts to improve the population throughout the search in order to create an optimal solution. Survival of the fittest technique also comes into the play whereby the system analyzes the parent population in order to create a list of activities that improve on the previous population. In GA method the activity list is also referred to as chromosome, where each unique activity is considered a strand of a gene. For a given list (Chromosome) the optimal solution can be achieved by employing S-SGS or P-SGS method. Diversification in the GA method is achieved by using the crossover method that identifies two or more crossover points in order to create a better sample. Intensification is incorporated in the model by using a different local search engine protocol that non-population based. If the search engine gets stuck in a non-optimal neighborhood the system employs the process called mutation in order to escape the non-optimal neighborhood. In mutation a single gene is chosen at random that is then switched with another gene as long as the switch does not compromise the feasible state of entire chromosome.

Under the GA method there are several different possibilities that are dependent upon the methods used to create new population sets. Mutation mechanism along with mutation strategy and parent chromosome heavily impacts the results of the method. GA method is considered to be very strong for global searches but is considered extremely weak in terms of local searches. GA has been found to produce the best results for RCPSP problems, some researchers have also tried to refine GA method by adopting path relinking method. Scatter search is a very good alternative to GA method that would be discussed later in the chapter.

Scatter search (SS) can be considered an alternative to GA method as they both employ the population and chromosomes in order to create optimal solutions (Laguna and Marti, 2012). The only distinction between GA and SS method is that the scatter search employs more diversification in its searches as compared to GA method. In SS method in addition to the chromosome a reference set of chromosomes is also used to create new populations. The reference chromosome is further divided into two subsets that contain the best solutions and diverse solutions. The new populations are generated by crossing pairs of first chromosome with the second set in order to achieve an optimal solution faster.

This approach is employed throughout the scatter search and is also called two tiered design. In order to incorporate enough diversification in the system the search engine ensures that the optimal solution is kept complete distinct from the diversification solution. A good SS search clearly identifies instructions in order to create initial pools of chromosomes that would be used in the search to create further populations. The improvement method used in scatter search usually employs double justification method, the reference material in order to ensure that new generations of chromosomes are created and the combination method to define the appropriate crossover techniques. Scatter search is also considered to be one of the most optimal search techniques in RCPSP.

Electromagnetism-Like is another alternative to GA search that also employs populations and chromosomes (Birbil and Fang, 2003). EM search engine is inspired by the Inverse Square Law. This law defines how the electrically charges particles interact with each other. The force between the electronically charged particles is considered to be inversely proportional to the square of distance between different particles. EM uses that principle to create population sets whereby each particle (activity) is considered to carry a charge that helps it either attract or dispel other particles from itself. By employing this technique the researchers are able to improve on the previous population results creating a viable search method.

2.6 Simulated Annealing

Simulated Annealing is inspired by the metallurgy techniques used to harden the metal (Dowsland and Thompson, 2012). The model maintains an optimal solution at all times which is compared with other solutions and when a new optimal solution is discovered the current one is discarded in favor of the new solution. In certain cases the system may adopt a worse solution as the current solution in order to escape from the non-optimal neighborhood. If the probability of the solution is less than a randomly selected number, then the system would choose a worse solution. SA method places special attention to the objective function of current solution and the new solution. The higher difference is the higher the probability would be and chances of accepting a worse solution. The model also uses a temperature parameter that impacts the level of acceptance of higher probability. A cooling system akin to metallurgy is also employed where a higher value gradually decreases towards lower probabilities. A higher temperature value encourages the mode to select a worse solution thereby inculcating diversification in the system. Higher temperature value forces the SA search engine to move to different neighborhoods improving the global search while a lower value favors the current neighborhood improving the local search. A good SA algorithm would contain means to ensure that the search remains in the current neighborhood yet provide means to jump to other neighborhoods in order to discover an optimal solution. SA search technique is considered to be better at local search as compared to global search.

2.7 Tabu Search

Tabu Search relies on the memory in order to detect the better solutions from the past experiences (Glover and Laguna, 2013). Memory is a collection of different past solutions. The system creates different types of lists are maintained in order to search different solutions. A short-term tabu (memory) is used to compare solutions in order to avoid being stuck in the same neighborhood. If the system is able to find optimal solution in the current neighborhood then it implies that search engine should not move to other neighborhoods but employ a different starting point in order to search new neighborhoods. The model also retains a list of poor solutions that provide an indication for the search engine to avoid neighborhoods that contain weak or poor solutions. Another area of the memory is dedicated for good solutions that provide an indication which neighborhoods should be explored further. Neighborhood search is similar to that of SA search and also incorporates guidelines in order to inculcate diversification and intensification in the search.

2.8 Ant colony Optimization

Ants use pheromone in order to identify the shortest route to the food (Mohan and Baskaran, 2012). Initially different groups travel different routes to reach the food and as the group with shorter distance makes more trips the pheromone on their path becomes more intense more quickly that signals to the other ants which patch is the shortest means to reach the food. In an RCPSP situation an ant select an order of activities the ant uses different strategic guides in order to determine the feasibility and effectiveness of the solution. These ants then collaborate to discover a set of good rules and create an optimal solution. Some of the measures in the system provide strategies to devise different solutions. The system escapes the worse neighborhoods by accepting worse solutions after a certain period of time has elapsed.

2.9 Artificial Bee Colony Optimization

Artificial bee colony is a recently developed technique that has gained immense popularity due to its simplicity, flexibility, acceptance and ability to create outstanding results (Karaboga et al, 2014). Bees employ a large number of techniques in order to find food and direct other members of the swarm to reach that food. ABC techniques employs three distinct group of bees called employed bees, scouts and onlookers. The bee directing the swarm to the food source is called onlooker bee while the bee that constantly visits different location to discover food is called employer bee and a bee carrying out different searches at random is called a scout. For each source there is one employed bee after its food source is depleted it becomes a scout again in order to search for a food source closest to its location.

2.10 Swarm Intelligence

Social insects or the insects that operate in group tend to be very good at searching and foraging food giving them distinct abilities to operate on their own with minimum inputs (Zhang et al , 2014). Such interactions are considered to be primitive but are very effective in achieving goals. This mentality has shown outstanding results in improving the computation abilities of the modern systems. SI systems are considered to be very effective in searching solutions for the distributed problems. This system is primarily inspired by bees, ants, termites and wasps.

2.11 Differentiation evolution

DE is a population inspired technique similar to GA (Bullough, 2013). It employs similar mutation, crossover and population algorithms. DE uses the mutation as a means to search solutions and guide towards optimal neighborhoods. DE uses a variable crossover that creates higher chances of accepting child vectors from the parents as compared to other population choices. By using the components from the existing population the system is able to create unique trial vectors. And then uses the crossover feature to shuffle information and create optimal solutions.

2.12 Uncertainty and Risk

Uncertainty models are used to convert the inputs to outputs (Rossi and Deutsch, 2014). Mistaking the uncertainty models and less important is the one of the most common mistakes that occurs in the modern models. Therefore it’s of vital importance to incorporate uncertainty measure into a model that would increase the efficiency of the overall results. There are two broad types of uncertainties called aleatory uncertainty and epistemic uncertainty. Epistemic uncertainty is reducible and is more subjective in nature, while on the other hand aleatory uncertainty cannot be reduced and is more random in nature.

Researchers have been using different models in order to create a good model for uncertainty. The most acceptable models employed include Probability theory, possibility, fuzzy set and Dempster Shafer theory. Each of the model identified tries to incorporate uncertainty in the system by analyzing the level of information available and incorporating a mathematical representation of uncertainty in it. A certain model should not be used if its application requires higher information in order to successfully implement then the current ones available.

It is extremely difficult to understand the events and the likelihood of their occurrence. If an event is expected to occur multiple times then its frequency can be expressed in per day or per year or other similar terms. But if the event is expected to occur only once, then its frequency is expressed through probability. If an event it expected to occur frequently but it's impossible or difficult to attach a numerical value to its occurrence then its frequency would be the probability of the frequency. The probabilistic risk assessment makes it compulsory to identify the information about all the occurrences. In case of the absence of the information the assumption that all the occurrences have equal likelihood should be used. This assumption is taken due to the axiom of addition that states that likelihood of all occurrences should be equal to 0 for example if probability of event a happening is .25 then the probability of event a not occurring should be .75.

This is a very rigid estimation regarding the likelihood of a threat materializing. In case of absence of information the probability should not be introduced because it could create a serious biasness in the model regarding the likelihood of its occurrence. Where the information is completely attainable and available in such scenarios probability theory is known to present the perfect results. Threat likelihood defines the probability of an event happening in the future based on the current or past circumstances. A higher degree of possibility or a lower degree of possibility does not necessarily imply that the event would have higher probability of occurrence or lower. But a high or low probability does define the possibility of an event occurring or not occurring.

Some threats would have a probability of zero attached to them but they should not be discounted because their occurrence could still materialize. On the other hand if an event has never happened that it is highly likely that it would not materialize in the future but that does not mean that it cannot occur. Fuzzy logic models have been extensively used in analyzing the information that is vague or ambiguous. But the fuzzy logic cannot deal with incomplete or inaccurate information. Ignorance cannot be analyzed by any means. Vagueness is defines as an uncertainty of a known event that is clearly measurable. All of these different methods provide a great solution for incorporating different models of uncertainty that would further improve the efficiency and reliability of a model.

2.13 Multi-Project Scheduling

Traditionally project management was only concerned about the management of individual projects, but in the real world circumstances nearly all projects are carried out in sync or combination with other projects (Caniels and Bakens, 2015). Multiple Project Management is a new trend in the modern business community. Despite a large number of organizations working with multiple projects there is very little concrete data available regarding the effective management of such projects (Spalek, 2012). Most projects follow the same cost and staffing requirements over the span of their life. Initially the projects tend to require lower budget and staff, then later on the cost and staff requirements start increasing drastically as the demand of the product increases in the market and drops with same speed in its final years (Meredith and Mantel, 2011). But, when it comes to the human resource and individual participants, their energy level, motivation and focus change over the life of the project. Using a common human resource on different projects increases the efficiency by eliminating idle time and the project also benefits from common expertise, improved communication and lower development costs.

It becomes extremely difficult to manage the allocation and consumption of resources in a multi-project environment. Resource management is the biggest issue that managers have to face in case of a multi-project management scenario. Human resource is one of the major issues that companies have to deal with in case of a project management, in addition to being the biggest cost in the project it is also the more complex. There are numerous methods described by management Gurus, but very little is known about which application or method is more suitable in which conditions, therefore more work needs to be done in this area to determine the effective techniques of dealing with the allocation and management of human resource. The purpose of this thesis is to evaluate different methods of resource allocation in terms of their efficiency and application in different PM scenarios.

There are several jargons that are used interchangeable in the world of project management, since different countries and disciplines have different definition for these terms, we would be using the guidelines and definitions outlined by PMI standards to distinguish them (Heagney, 2012). Portfolio refers to a collection of different programs, projects or other type of jobs merged or categorized together for the purpose effective management and ensuring coordination. A program is defined at a group of activities managed together in order to attain benefits and rewards in a coordinated fashion and these activities if managed individually would not results in a controlled or effective allocation of resources. In order to avoid confusion, we would be relying on the term of “Multi-project”.

Resource allocation and management could have several objectives and goals, but in this thesis we would be focusing on attaining the tactical advantage for the business through effective resource allocation and scheduling. Tactical scheduling refers to the management of a project in such a way that the required activities could be performed on required time with the needed resources. When referring to resource, we mostly mean Human resource. Multi-project management scenarios with limited resources can be dealt with using different methods and techniques. The issue of Multi-project is dealt with by creating artificial project activity nodes at the beginning and end of multi as well as single projects. Therefore the multi-projects are treated in similar manner as a large project with multiple sub-projects or activities. Using this assumption it should be possible to resolve the issue of multi-project management using the individual PM methods such as Exact Method, Meta-heuristic and Heuristic Methods. Since the multi-projects have several activity nodes they are classified under a unique category called Network Optimization Algorithm (Heagney, 2012). PM scenarios always carry a lot of embedded uncertainty in them. Algorithms to deal with uncertainty that are primarily used in single project management scenarios can be used as reference or beginning points for developing effective solutions for multi-project management scenarios. Some of the methods commonly used include Reactive Scheduling, Fuzzy Project Scheduling, Sensitivity Analysis and Stochastic Scheduling. Critical chain scheduling is a proactive approach towards dealing with

Multi-project scheduling issues, its effectiveness primarily relies on the application of Theory of Constraint and Capacity constraining resources (Leach, 2014).

Many aspects of the project management rely heavily on the individual's ability, skills and prior knowledge. Different tasks have their own unique characteristics and require a set of different skills to efficiently deal with them. Several researches under the banner of Process Enactment are concerned in this very direction (Weske, 2012). This thesis propose a separate category called Resource modeling and Simulation to collect and allocate rigid resource definitions, policy driven resource allotment and continuous monitoring and simulation of the different states of resources. The multi-agent planning and resource scheduling methods are classified under a special category due to its distributed algorithm nature.

Many of the activities in a project rely heavily on the people requiring proactive and instant responses for an effective Project management. Proactive methods of management place a vital importance on the application in real life situations when the project is continued with a continuous iteration. Proactive or agile practices can be extrapolated in order to deal with larger of Multi-project management scenarios. A sharing of human resources in terms of technicality and leadership can be used as a renewable resource from different teams or projects. Proactive approaches dictate that projects of different priorities should be dealt by the people of different competencies such as unattractive or difficult project should be handled by different teams in turns rather than attaching the entire management responsibility to a single individual.

Some managers tend to increase the size of a smaller project in order to incorporate multiple objective and products in it. The enlarged team could face the situation of too many cooks and compromise the effective management of the project. But in companies with smaller size or limited resources it can be very effective as a single team of programmers, network managers and business experts can deal with multiple smaller projects under the leadership of a single individual. Usually the members in a team are divided in 3 different groups, All-around members, service employees and individuals possessing expertise in a single field (Experts). Different project scheduling patterns and methods could be used to deal with different kinds of resources.

2.14 Aspects of Management Scenarios

Different multi-project management suits different management and organizational structures (Aaltonen, 2011). There several aspects which can be used to measure different methods in a systematic and efficient manner. Very first aspect is the value orientation. How a method deals with categorization and management of different resources leads to varied approaches to project rescheduling methods. Different questions like how the utilization and cost interact with each other, the nature and quality of the resource all of these aspects are heavily dependent upon the requirements of the project and the policies of the management.

The question of centralization also heavily impacts the PM techniques. The centralized or autonomous allocation can have different impacts upon the project. Centralized or decentralized allocations have different costing systems as well as the structure. In some projects the intensity of centralization or decentralization can be changed based on the requirements, while other time the structure and management style of the company cause hindrance to the changes in the centralization or decentralization allocation of resources.

Another importance aspect is the uniqueness of the project, more unique characteristics multi-projects possess; more difficult it would be to treat them as a single large project. Multi-projects activities and resources to have some common characteristics and requirements that enables the management to deal effectively with multi-project management scenarios. Similar characteristics within different projects make it easier to adopt similar approaches and resource allocations. The similarities of different projects may present themselves as having identical/similar objectives, similar business environment, customers with similar preferences or tastes, similar legal concerns, identical location and interchangeable skill sets or technologies. A single project can have different kinds of similar characteristics requiring different approaches to the management.

Risk, uncertainty and complexity are also very essential aspects of a management scenario. These components are very easy to distinguish and understand as they serve the main indicators of differences between the traditional and proactive or agile project management conditions. Besides the common complexities of the projects (Size, location and knowledge gaps) other impactful factors contributing to the complexity of the project include the level of interdependence between different projects. Different methods would demand different level of costs and management capabilities. The existence of certain organizational structure, tools or the skills of participants may be prerequisites for the application of certain methods of management. The culture of the organization, management style and skills can also contribute towards the selection of one management method over another.

2.14.1 Value Orientation

The production costs involved in the project can be substantially reduced by sharing resources amongst different projects (Yaghootkar and Gil, 2012). But, sharing may lead to technical compromises or place heavy limitations over the operation and creativity in the team. Many project management techniques state that it is essential to focus on a single project or goal in order to effectively manage a project. Therefore individual Project management scenarios encourage application of creativity rather than consumption of resources. Companies can achieve greater quality by sacrificing dedicated or unutilized resources. But most of the times the resources are very limited in the companies and the allocation and consumption of resources is considered to be the beginning point in most Multi-project scheduling techniques. Networks can select different optimization objectives or priority rules in form of minimum lead, resource consumption to the maximum limit or minimum slack acceptable in a project. But is not recommended to deal with individuals as machines.

Critical chain bases its management on the Theory of constraint to increase the overall throughput of the entire system rather than picking and choosing certain resources (Leach, 2014). Resource modeling refers to the delegation of right team and people to appropriate tasks in order to increase the accuracy and effectiveness, which places far more importance over the individual’s skills and experience. Multi-agents are used for a more adaptive decision making scenario. Agile or proactive methods tend to place more trust over the communication abilities of the teams and individuals. Common Agile methods include Virtual team, Pair programming and core teams. Different methods have their own requirements and considerations that the management must be wary of.

2.14.2 Centralization

A frequent misconception about Multi-project is that it’s supposed to be a very large project, but in reality it does not have to be. If management treats multi-project as a single large project and applies the methods used for the management of single projects then it would eventually lead to: an unrelenting cycle of bureaucratic policies, a linear project lifecycle that lacks any kind of flexibility to react towards the different changes that occur in a dynamic environment, and numerous repetitive technologies. Coordinating different projects together does not eliminate the need for the integration of the functional and critical areas of the individual project, but rather increases the level of focus and complexity of the coordination among the projects. Cross Functional Synchronization and project management become necessary for efficient project management.

Individuals projects tend to allocate resources centrally before the inception of the project and have little disruptions after they have been carried out. Critical chain, Network and Multi Project are essential methods of allocating resources across different activities of the project in their essence. They tend to put less stress over the fundamental purpose of the creation of a project. Generally if the relevance of a team between different projects is weak then the project team should be left relatively constant, relying on the scheduling as a primary technique to deal with uncertainty and training. Different heuristic techniques can be used to instill a level of decentralization in these techniques and methods. For example in a group multi-agents tend to created fixed set of agents and encourages a dynamic allocation of marketable agents throughout the entire group. Agile methods on the other hand tend to encourage a more autonomous approach of scheduling to each project and incorporate an adjusting point to reconcile the differences. Classification has a very high reliance over centralization and relies on proven and stable methods of allocation for each resource type. The federal modes of managements have to contend with a lack of central objective for senior management in case of multi-projects.

2.14.3Homogeneity

Homogeneous or similar projects make it easier for the management to adopt the common project management approaches. By using these common methods the management can create a level of consistency amongst the different projects and enable them to formulate common reports regarding the projects. Common methods propagate coordination and also facilitate the resource sharing amongst different projects. Nearly all methods encourage the management to group the project together that possess similar characteristics. But in real life certain situations may arise where the projects tend to be difficult to group due to the dissimilarity. It is more advisable to use common and stable methods on multi-projects rather than relying on different approaches.

In case of the management of project management no similarity needs to present amongst the projects. Critical chains must have the need of common resources to properly link the project together. Resource modelers, Multi-agents and Networks demand identical or similar tools in order to run transferrable date. Multi-projects that are strong related to each other or try to achieve a common goal should be schedules centrally and treated like a large project to improve coordination. It is highly recommended to group the projects together that possess similar characteristics otherwise the less important projects would be deprived of the resources as all resources are confirmed on larger and more important projects. Resource modeling techniques can allocate and schedule resources by exchanging corresponding resources. The principle similarly applies to the management of single projects as well.

If the projects possess certain common characteristics such as similar scope, requirement of common technology and the room for employing reusable components then it is more logical to use Agile methods of project management. Some of the agile methods that fulfill this purpose include Virtual team, Pair programming and Core teams. When all is said and done the main purpose of project rescheduling activities is to facilitate the sharing of common resources amongst the project. Pair programming is very technical method and certain group advocate against its efficient, but in the long run it can be very useful in expert of Novice cases. A specialized resource can be paired with the stable members of the project in order to transfer the expert skills and shorten the waiting time before the resource can be reassigned to other projects as well. Newer employees can learn quicker if they are paired with the old or stable members of the project. It can also facilitate the sharing of resources amongst uncommon or heterogenic projects. Multi-product is an uncommon practice but may prove to be handy in certain cases such as product line, small organization or specialized tasks.

2.14.4Complexity

Singular project management can assist in providing solutions for highly complex or commonly managed platforms. A large project may carry higher risk, therefore it should be broken into smaller projects to assist in reducing the level of uncertainty and risk in the project. Networks are employed to provide solutions for extremely complex, but limited resource project scheduling issues and some algorithms intended to optimize the solution tend to crash when applied over a large activity network. The result formulated through Network schedule tend to add a specific price to switching as well as communication activities that can be disproportional with high threshold. Critical chains are much more effective in dealing with the projects that have higher complexity due to their size or amount. But these issues can be simplified by using a diverse utilization rate or use different quality and type of resources.

It is difficult to run Resource modeling and other models such as Multi-agent through automatic algorithms, but they have the ability to simplify, simulate and deal with complex resource scheduling situations. Critics state that the Agile Methods tend to lack the ability to apply to larger projects due to their size limitations. The higher complexity in the larger projects can leave the Agile Methods to become ineffective and prove wrong outcomes. Core Team, Multi-project and Pair Programming have only been applied to the smaller or medium sized companies and their utility in large or complex projects remain untested. These agile methods may be combined with other models to account for the complexity or size of the larger projects. Differentiated Matrix method has proven to be very effective for larger companies and it is very similar in nature to the Virtual team method.

2.14.5Uncertainty

In cases where the resource scheduling is not applied to the individual projects, it tends to create a disparity by failing to incorporate coping mechanism with uncertainty. In such situation a buffer resource should be maintained to deal with any uncertainty. For Network to be effective it requires perfect set of information that can be very tricky to find in real world. When the projects fail to meet the budgeted timeline or lag behind the planned schedule then it becomes impossible to activate scheduling of other projects. Despite the existence of certain methods to deal with uncertainty such as buffer resource the still carry a very critical weakness by assuming that the priority and the order of tasks have been defined in the beginning which is highly unlikely. When certain changes materialize it becomes essential for the networking to be changed and restructured accordingly. Multi-agent and Common Modeling techniques have better coping mechanism for dynamic resource allocation but tends to follow the same scheduling principle as Network method. The main purpose for the creation of agile methods is to deal with the uncertainty. Agile methods tend to use repetitive allocating and scheduling methods. The core team containing the engineers and programmers need to be fixed on a project to ensure stability while the supporting staff can be exchanged through multiple projects in order to deal with the uncertainties. Pair programming is a fair method of transferring support staff amongst different projects. Virtual teams are also very effective in meeting the constraints by allocating some-time of their project to deal with uncertainty. Exchange of resources amongst different projects can be achieved through Core-team when the need arises. Categorization should be based upon the guidelines specified in different categories.

2.14.6Executive ability

Individual project management techniques tend to rely heavily on the required quantity of resources and employ documents to facilitate communication with similar or related projects. In Network, Multi-agent and Resource modelling it becomes extremely difficult to handle large amounts of data manually without the assistance of computer generated programs. These methods require an existence of integrated system and standardized processes to acquire, update and disperse data. Resource modeling requires the details about the skills, classification and requirements of different resources to be truly effective. In Critical chains it is considered to be effective technique to maximize throughput even if only a single resource is fully utilized across different projects. The co-operation and decision making abilities of the resources are considered to be a pre-requisite for all kinds of agile methods. Let’s prove this by taking the example of network method: if a project manager fails to communicate the priorities of the project through the ordinary channels then he may take unrealistic or drastic measures to acquire resources for his project. In several agile methods the scheduling of the resources is ideally based on the negotiation between the parties and co-operation plays a very critical role in success. Managers may be resistant to lose resources under the belief that the resources allocated to other projects may not be returned to them when the need arises. It is highly recommended to repetitively transfer the resources except for the experts or stable resources and avoid any complexity that might arise through other methods. By synchronizing the control points of related project this can be effectively achieved. While sharing resources it is advisable to concentrate on a single product rather than stressing the company's resources thin over multiple products.

2.15 RCMPSP vs RCPSP

Each project has a series of activities that use the multiple types of resources from a common pool that tend to be smaller than the requirement dictates for all projects to utilize their resources simultaneously. Therefore the rules should be established that propagate effective sharing of resources in order to achieve a certain objective such as minimizing the idle time for projects before they can get the required resources. That is the main issue faced in a RCMPSP scenario. Most of the research has been centered on the management of single projects. Commonly 2 different techniques can be used by the managers to deal with multi-project scenarios (1) a modified single project approach that uses dummy activities in order to reconcile different projects under a single large project which would result in the limitation of RCMPSP to a single pathed RCPSP, (2) Multi-project approach that maintains the RCMPSP and creates a separate critical path for each project in the portfolio.

RCPSP does not possess the ability to effectively find solution in polynomial time. The same weakness applies to RCMPSP as well. Therefore most researches recommend the application of heuristic or meta-heuristic techniques. PR or Priority-rule Heuristic techniques have a very important place for several reasons: (1) Meta-heuristic solutions tend to require greater computation capability creating a need for PR to deal with large issues, (2) PR are a necessary part of other heuristic techniques and cannot be dispensed for formulating solutions for other meta-heuristic solutions. (3) PR are very popular in commercial projects due to their innate simplicity and speed. However, the biggest argument in the PR's favor is that they are essential in the practice. For multiple reasons it is impossible for the managers to create activity networks that rely on Meta-heuristic techniques. In real life managers tend to make quick decisions based on their experience rather than running a complex simulation based on meta-heuristic techniques. Therefore most of the information available in the textbook does not provide realistic methods to formulate concrete system for the managers. This issue of indecisiveness arises because of the limited testing of common PRs in the literature that fail to deal with all possible situations that could arise. In addition to that the literature also provides conflicting results of different PR due to several variables used in the calculation. Therefore there is a very strong need and demand for firmed guidelines about PR and PCMPSP. While there are numerous theories about a perfect way to address RCPSP issues in a single project, the approach tends to have a lot of drawbacks when it comes to solving RCMPSPS. The biggest weakness in this approach is that it makes several unrealistic assumptions which do not hold true in the real situations, such as the assumption that projects have equal delay penalties. Another major hurdle is the difficulty to manage different projects under a single multi-project, in reality each project has its distinct features and has a dedicated manager who is solely concerned about the success of his project.

MP approach highlights two approaches to deal with the management, these are called Exact and heuristic approaches. Exact methods are ideal for dealing with smaller problems but tend to be ineffective in larger RCMPSP situations. Heuristic approaches are generally divided in form of 4 different categories called PR Based, X-pass Heuristic and miscellaneous heuristics. Classical heuristic approaches tend to rely heavily on simulated annealing and swarm optimization. While Non-swarm tend to favor agent based and Population based approaches.

PR based heuristics include single and multi-pass methods. Single pass put more importance on the activity that achieves optimization for a certain value. Multi-pass approach inculcates different priority rules and tend to employ several PRs one after another and has a large degree of randomness it in. PRs are usually distinguished by the type of information that they require: (a) Activity based, (b) project-driven, or (c) resource based. Activity-based put more importance on the characteristics of the activity itself. While project PRs tend to favor those activities that possess similar characteristics as the projects that they belong to.

It has been established that highly successful PRs tend to inculcate some form of time and resource usage measure. The researchers have also identified three essential characteristics in RCPSP scenarios: How much resource does an activity need or utilize, the average slack that would be experienced by the activity when designing critical path length, and the complexity involved in the project. Number of activities in a project is not considered to be highly relevant when determine the success of a project. Network Complexity, resource utilization factor and successful completion of critical activities are considered to be the most important indicators when determining the success of a project’s performance.

There are a horde of studies that have been conducted over the success, effectiveness and reliability of different PR in RCPSP scenarios, only a very limited progress have been made in a Multi-project management environment. It is critical to understand that single and multi-projects scenarios have different outcomes under similar PRs. Single project management is considered to be highly effective in minimizing duration of a single-project, while multi-projects tend to perform better in reducing the average time required in multi-project scenario. RCMPSP studies have failed to formulate a single result about the performance of different PRs under different conditions. Multiple results and lack of single rhetoric has led to lack of clear guidelines that managers can use when it comes to the application of different PRs.

2.16 RCMPSP characteristics and Measures

There are four main characteristics for successful optimization of RCMPSP. These include Network complexity, resource contention and resource distribution. These four aspects are considered to be very important in proper resolution of projects in different situations.

2.16.1 Objective Function

There are several different kinds of objective functions that managers rely upon for RCMPSP. The most popular measure is minimization of project duration. In addition to this measure there are other different kinds of objective functions used by the managers including total project delay, overall project cost and minimization of project cost. Studies have confirmed that the PR performance is heavily influenced by different objectives that are used. For example in order to measure delay, the managers specify the project's due date and any period used over this date is considered to be a delay. The issue of delay can be measured through 5 different ways:

1. Total Delay during the project or activity
2. Average delay
3. Percentage of Average Delay
4. Maximum delay encountered during a project or activity
5. Percentage of maximum delay

The first three measures put higher stress on the project measures while the later seek to identify the portfolio problems. First two measures seek to discriminate between two equal values. The percentage delay provides a comparable measure that can be used to measure the success or failure of the project against similar projects. Measure 4 and 5 incorporate the personal views of the managers dealing with the project. It is the discretion of a manager about what he thinks would be the best value of measurement.

2.16.2Network Complexity

Networks with lesser complexity tend to be less constrained as compared to the alternatives. It is suggested that an adaptive number form should be used by the managers to normalize the complexity of the network. In a multi-project a composite measures should not be employed as there is very little evidence regarding their effectiveness of ineffectiveness in such situations. Three complexity issue may not be same as a one high or two lower complexity problem despite the fact that both provide similar averages. Therefore it is important to incorporate or maintain the distinctive features of the projects and employ a vector of constituent to deal with MP.

2.16.3 Resource Distribution

Researchers have developed several different measures to distribute the resources and determine their availability. The most commonly used measures include resource factor (RF), a measure indicative of an average number of activities or resources utilized, and (RS) resource strength that expresses the requirement of the activities and availability of resource against such requirements. However these resource distribution measures may not be as effective in a Multi-project environment and require a distinct measure in RCMPSP cases. ARLF (Average Resource Loading Factor) that seeks to identify whether the total resources of the project are saved at the front of back of the constraint chain. The ARLF tends to have several critical problems, first it that it can become biased and lose its ability to distinguish between different cases when percentages are used. Second is that despite its widespread application in the MP environment it has been primarily created for single projects. A normalization ARLF called (NARLF) is a modified version of ARLF that can be used to counter these problems.

2.16.4 Resource Contention

UF (Utilization Factor) to deal with the resource contention issues. UF is calculated by comparing the resource required with resource availability in a specific period. If the UF is estimated at less than 1 then it is assumed that there is no resource contention amongst the projects. An average UF is also used to decrease the computational intensity between different intervals.

2.17 Multi-mode RCPSP

In multi-modes, the modes provide different alternative combination and quantities to complete certain activities. For example in certain situation an activity can reach its conclusion faster if more quantity of a resource is applies to it called Time-resource trade. These methods were initially identified for single RCPSP. In reality most of the algorithms in exact condition tend to apply insinuated inventories with bound and branch. There are several different types of heuristic and meta-heuristic to deal with scaling instances. GA is considered to be very effective to deal with large scale issues due to its characteristics.

While on the other hand some researchers recommend Multi-mode RCPSP that are based on genetic algorithm techniques to inculcate a time-cost mode of trade-off. These modes employ several critical matters that have been ignored by other researches. These critical factors are:

- Taking account of indirect costs along with indirect costs of a project
- Challenging the assumption of limitless renewable resources
- The performance of each activity is closely linked to its selected mode

Mode is a method through the time on an activity is reduced by employing higher direct costs.

2.18 Chapter Summary

This chapter provided an extensive literature review for the issues discussed in this thesis. First, it discussed resources scheduling problem in project managements and heuristic approaches used to addressed. Then, it went to elaborate on non-conventional meta-heuristics approaches which are used by many in literature to increase effectiveness of scheduling process. Lastly, this chapter highlighted different aspects of scheduling problem such as complexity which is need to be considered when developing any scheduling solution.

Chapter 3: Methodology

3.1 Introduction

This research is adopting the use of Genetic Algorithm to find near optimal solutions for resource scheduling problems. The main purpose of this chapter is to illustrate how this research is implementing Genetic Algorithm in context of project management. Since Genetic Algorithm is a kind of evolution algorithms family, it is beneficial to discuss Genetic Algorithm based on its evolutionary aspects to explain the justification for its effectiveness.

3.2 Theoretical Framework

This research tries to investigate resources scheduling in multi project environment. It highlights their recent advances in literature regarding this issue. Also, it is studying the possibility of using mathematical programming and artificial intelligence techniques to solve this problem. Resources scheduling in project management is one of the most important issues that faces managers around the globe. The complexity of this problem has been increased drastically due to the introduction of project portfolio management. Numerous researches have concluded that usually project managers manage multiple projects at the same time. Normally, in environment where multiple projects are being managed at the same time, the available sources are commonly shared. The impact of pulling resources from one project to another may have drastic effects on overall projects performance.

The first step to investigate the existing theoretical framework is to give a simple definition about projects. In a nutshell, a project is a set of several tasks. Every project has an objective. This objective is achieved by the end of the execution of the final task. In general, the objective is to produce new products or services in attempt to achieve profits or some kind of benefits. These project tasks have a predefined order of their executions. Some tasks are executed before others; while other tasks may be executed concurrently. Each one of these tasks requires some sort of resources. Usually, resources are very limited. Also, they are shared between tasks. As a result, some tasks may be delayed because there are not enough resources to complete them. Keep in mind, some resources may be renewable; while others are not. For example, renewable resources can be human resources. An employee performing a task can be asked to perform another task after the first one is finished. An example of non-renewable resources can be money or raw materials.

To illustrate how the execution of tasks may be effected by limited resource availability, we will use the following example. Imagine that we have project A. This project has multiple tasks. Each task has an index to indicate its order. For simplicity, we will use this following straightforward order. Each task has an index (j). If this task index is an odd number, then this task will be executed directly after the previous two tasks (task j-1 and task j-2). If the task index is an even number, then this task would be executed after the even previous task execution (only task j-2). The following diagram depicts this execution order.

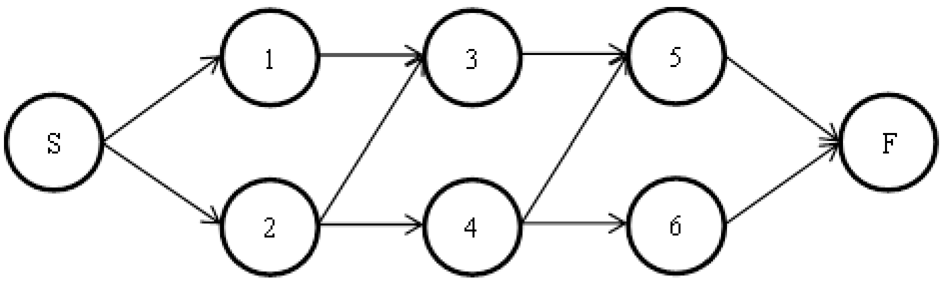


Figure 3.1: Simple project tasks execution order.

Keep in mind that in reality any column in figure 3.1 can have any number of tasks. Similarly, any row can have any number of tasks. Also, the connection between tasks can be of any sort. This example of execution order can allow some tasks be to be executed concurrently (tasks in the same column) if their previous tasks have been concluded. Each one of these tasks requires resources. In general, these resources are different for each task. However, some tasks may have the sane resource dependency. If the shared resources are very limited and these tasks are executed at the same time then these resources should be scheduled in a way that advances the project in the most efficient manner.

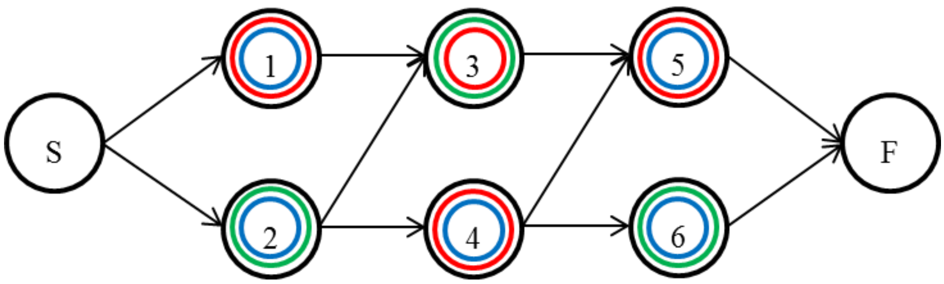


Figure 3.2: Simple project tasks execution order with resource dependency.

To expand the example in figure 3.1, three resources types were introduced. For simplicity, assume that there is only one unit of each resource. The dependency of tasks on these resources was depicted in figure 3.2 by using colour code. It's a clearer that because of this resource dependency some part of execution order should be changed. For example, task 3 and task 4 cannot be executed at the same time because both of them depend on RED resources. As a result, one of them should be executed before the other while the other is waiting. The same situation can be seen in every column. In this example, due to the very limited availability of resources, there are no tasks which can be executed concurrently. Imagine that the manager who is managing this project was asked to manage another project (project B) which is similar to project A. Here, a multi project environment is created where two projects are sharing the same resources. As a result, both of projects would be affected and delayed depending on the resources scheduling policy adopted by the manager.

Now, imagine that the available amount of resource RED is increased to 2 units. In this case, task 3 and task 4 of the same project can be executed at the same time. Or, task 3 from project A and task 4 from project B can be executed in concurrently if we assume that both of projects are at the same progress point. Here, the manager has to decide if it is more beneficial to execute tasks from different projects or execute tasks at the same project. The later decision is totally depending on how each of projects is prioritized. Also, it is depending in on how each project objective is defined as well.

3.2.1Formal Definition

This section tries to ease dealing with the resources scheduling problem by defining it formally. Each project has M number of tasks. These tasks are indexed as follow, T_i where $i = 1,2,3, \ldots, M$. There are N number of resources where the j-th resource is indicated by R_j , $j = 1,2,3, \ldots, N$. Each i-th task requires r_{ij} amount of j-th resource. The amount of processing time for the i-th task is denoted by p_i. The allocation function A(T,R,t) is used to show how each resources is allocated to each task throughout projects progress time. This function takes three inputs. The first one is task indicator. Second is resource indicator and the third input is current time. If the i-th task is being executed, the allocation function should return r_{ij}. Otherwise, the allocation function should return zero. The total available amount of j-th resource is symbolized by $\bar{r}_j$. Any resources scheduling mechanism should guarantee the following constraint at any moment of time:

$$\sum_{i}^{M} \sum_{j}^{N} A(T_i, R_j, t) \leq \bar{r}_j , \forall\ i, j, t$$

This model can be extended by introducing execution mode concept. This concept states that there are several mode of task execution. Task processing time (p_i) and task required amount of resources (r_{ij}) are dependent on the execution mode. Let's denote the execution mode of the i-th task by m_i. To elaborate, executing a task under one mode may require smaller amount of resources and longer processing time; while executing the same task under another mode may lead to shorter processing time and more resource consumption. Note that each task has start time and finished time. Let's denote the start time of i-th task by s_i and finished time of i-th task by f_i. Any resources scheduling mechanism will generate two vectors as the scheduling decision. These vectors are the start time of all tasks $\boldsymbol{s} = \{s_1, s_2, \dots, s_M\}$ and the execution mode of all tasks $\boldsymbol{m} = \{m_1, m_2, \dots, m_M\}$. Keep in mind that due to the predefined execution order of project tasks, the start time of some tasks has to be larger than the finish time of others. These constraints have to be guaranteed during any resources scheduling process.

3.2.2Resources Scheduling Objectives

As mentioned before, resources scheduling in any project has to achieve some predefined objective. This resources scheduling objective can have many forms and definitions. According to Gomez et al (2014), most works in literature use these following objectives:

- Minimize project lifespan
- Minimize project cost
- Minimize the start time of project tasks
- Maximize project net present value

Let the finish time of the last projects task be f_{M+1} which is a dummy task. Then, we can formally write resources scheduling problem where the objective is to minimize project lifespan as follow:

$$\min_{\boldsymbol{s},\boldsymbol{m}} O(\boldsymbol{s},\boldsymbol{m}) = f_{M+1}$$

subject to:

$$\sum_{i}^{M}\sum_{j}^{N} A\left(T_i, R_j, t\right) \le \bar{r}_j \text{ ,} \forall\, i, j, t$$

$$s_i \ge f_j, where\ T_i\ depends\ on\ T_j$$

The first constraint in the previous problem guarantees that the distributed resources are less than or equal the available sources at each moment of time. The second constraint enforces execution order of projects task. For simplicity, let assume there is only one mode for each task to be executed. Therefore, resources scheduling mechanism will return only vector **s** after solving the previous optimization problem. Next step is to define an approach to differentiate between different scheduling decisions when multiple objectives are considered.

This thesis would adopt Pareto dominance concept to distinguish between different scheduling decisions. Let's imagine that we have two scheduling decisions $\boldsymbol{s}$ and $\boldsymbol{s}'$. Scheduling decision $\boldsymbol{s}$ dominates $\boldsymbol{s}'$ if $\boldsymbol{s}$ achieve better performance than $\boldsymbol{s}'$ for all the objectives. In other words, $\boldsymbol{s}'$ is dominated by $\boldsymbol{s}$. On the other hand, if $\boldsymbol{s}$ does not dominate $\boldsymbol{s}'$ in all objectives, then both of these resources scheduling decisions are indifferent. The Pareto optimal solution is the resources scheduling decision that outperforms all other resources scheduling decisions in all objectives. The last concept we need to know is Pareto optimal front which is the set of all Pareto optimal solutions. Normally, solutions in Pareto optimal front are indifferent resources scheduling decisions.

The work published by Gomez et al (2014) is one of the most recent researches on the topic. The authors investigated several multi-objectives resources scheduling techniques. The theories behind these techniques are based on meta-heuristic approaches. All of the proposed techniques use two important concepts. These are priority rules and solution neighbourhood. At the beginning of resources scheduling process, non-dominated solutions should be generated. The scheduler selects a set of tasks that can be executed directly after the first dummy start task. Which one of these tasks should be added to the resources scheduling decision first is decided by one of the priority rules. The first rule schedules tasks which have smaller processing time first. The second rule schedules tasks which have larger number of successor tasks first. The last rule is schedules tasks which have lower predefined weights first. Keep in mind, tasks which have been scheduled first will have the right to all available resources before later tasks. This scheduling process will keep progressing for the remaining tasks given that all constraints are met.

After generating resources scheduling decision, a neighborhood of this decision would be constructed. This neighborhood is a set of other resources scheduling decisions which are very similar to the decision generated from previous step. There are two ways to generate this neighborhood. Either by permuting two adjacent tasks in the original resources scheduling decision; or by moving a task to another position in the decision sequence. Note, all neighborhood solutions should guarantee that all constraints are met. At the end, the solution which has the best performance for all objectives would be chosen as the resources scheduling decision.

3.3 Research Deployment Process Overview

It is widely documented in the literature that there are five essential components of academic research. These are:

1. Objective and purpose of the research
2. Theoretical and Conceptual context of the research
3. Questions that drive the research agenda
4. Methods for deployment the research to answer the questions and accomplish the research goals
5. Research validity and robustness

The process used to deploy this research is illustrated in the following diagram.

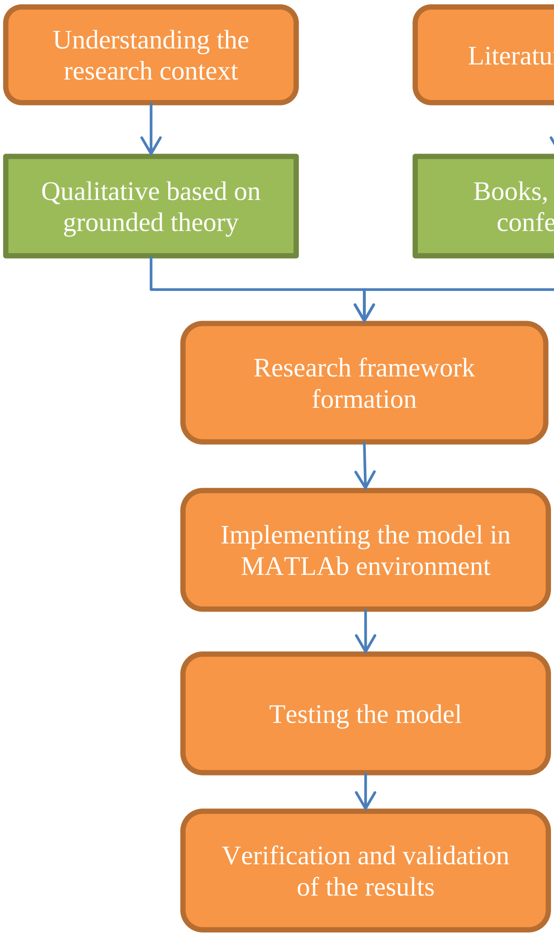


Figure 3.3: Research deployment framework.

3.3.2 Objective and Purpose of the Research

As mentioned before, the main purpose of this research is to address the issues raised in literature such as uncertainty in wider project management situations. The following are the main objective of this research:

- To tackle resource scheduling in project managements in a way that addresses shortcomings in literature especially uncertainty effect.
- To use advanced meta-heuristic approach such as genetic algorithm to build an automated resource scheduler under uncertainty.

- To evaluate and test proposed approach based on adopted standards in literature so that valid comparison can be established.
- To evaluate the proposed approach based on real resource scheduling task in real project management situation.

The deployment process of this research is evolved around achieving these objective in their entirety.

3.3.3Theoretical and Conceptual Context of the Research

A project is a set of several tasks. Every project has an objective. This objective is achieved by the end of the execution of the final task. In general, the objective is to produce new products or services in attempt to achieve profits or some kind of benefits. These project tasks have a predefined order of their executions. Some tasks are executed before others; while other tasks may be executed concurrently. Each one of these tasks requires some sort of resources. Usually, resources are very limited. Also, they are shared between tasks. As a result, some tasks may be delayed because there are not enough resources to complete them. Keep in mind, some resources may be renewable; while others are not. Scheduling of these resource is the main problem of this research.

3.3.4Questions that Drive the Research Agenda

Based on the established theoretical framework and conceptual context, this research attempt to answer the following question:

- To what extent the uncertainty problem affect scheduling process in project management?
- Can meta-heuristic approach such as genetic algorithm provide a better solution to the problem resource scheduler under uncertainty?

- What is the best approach to utilize meta-heuristic optimization techniques in performing scheduling process in project management?

Designing a deployment process for this research so that answers for these question can be figured out is very essential for this research.

3.3.5Methods for Deployment the Research

This research is divided into phases. Each one of these phases composes of a set of activities.

Analysis and Development Phase

As said before, this research will use qualitative and quantitative research approaches to develop good resources scheduling mechanisms for project portfolio management. Both of the analysis and development require a lot of work in term of gathering the needed knowledge which will be covered by the next phase.

Literature Review

Here, all the related works in literature should be reviewed, summarized and criticized. Some works may have a good solutions and insights which can be used in this research.

Proposed Theory Formation

After acquiring the necessary knowledge as a result of literature review, the researcher will be able to employ this knowledge to develop theoretical basis for the proposed resources scheduling mechanisms. Hypotheses on how resources scheduling mechanisms should operate would be proposed during this activity. The general algorithmic behaviour of the proposed resources scheduling mechanisms will be developed during this activity.

Modeling and Simulation

The proposed mechanism is modelled using simulation approach. The reason behind using this approach of testing is to investigate the abstract performance of the proposed mechanisms without the effects of other factors which are found in reality such as manager preferences.

Results Generation

An actual implementation of the proposed resources scheduling mechanisms using Matlab environment is performed. Every aspect of the proposed resources scheduling mechanisms can be traced and evaluated in very efficient way. A lot of these traced data will be converted into information that describes the evacuated system. Further investigation on the generated information led to forming this research results.

3.4 Research validity and robustness GA methodology Overview

It is evident from the history that evolution algorithms can take many forms and variants. Despite these different variants the guiding principle remains the same that a certain individual is forced to exist in an environment with a limited amount of resources and all the members of the population have to compete in order to gain those resources creating the scenario similar to the survivor of the fittest. An individual in the context of resource scheduling in project management can be considered as a possible scheduling decision. This competition in turn improves the strength and fitness scale of the individual participants in the population. Similarly such analogy could be considered resource scheduling In other words, several scheduling decisions (individuals) form a population. Then, a competition mechanism is placed in the population to force individuals (scheduling decisions) to improve their fitness. Based on this analysis it is possible to maximize certain traits or qualities in the population (group of scheduling decisions) that can be utilized to create a solution from the population. This quality function is then further extended to the algorithm in order to create an abstract type of fitness measure. Keep in mind, genetic algorithm is a generic technique. User of GA has to provide his own quality function. There are no default quality functions for all uses of GA. The quality function in this research can be related to project management such as Project Lifespan. The generated fitness values then form the basis for the creation of the next generation; the algorithm uses the fitness values to seed the individuals in the next generation.

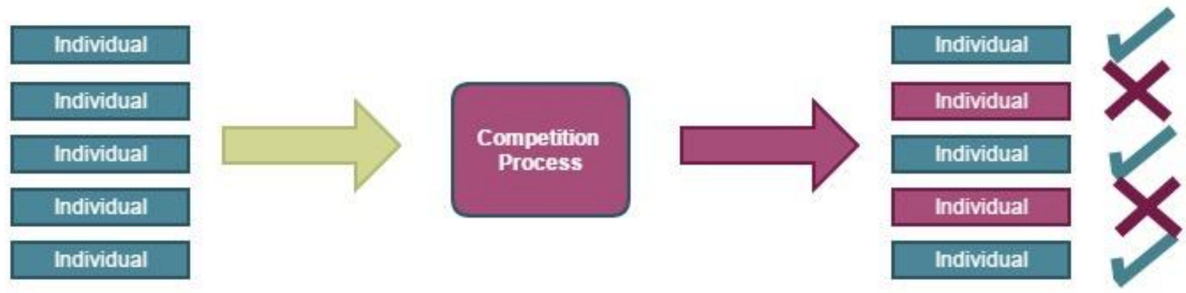


Figure 3.4: Survivor of the fittest.

This process is accomplished by applying the measures of mutation, crossover or a combination of both. In crossover, an operator is applied to at least two or multiple individual candidates (That become the parents of the next generation), resulting in the production of new candidates (Children).

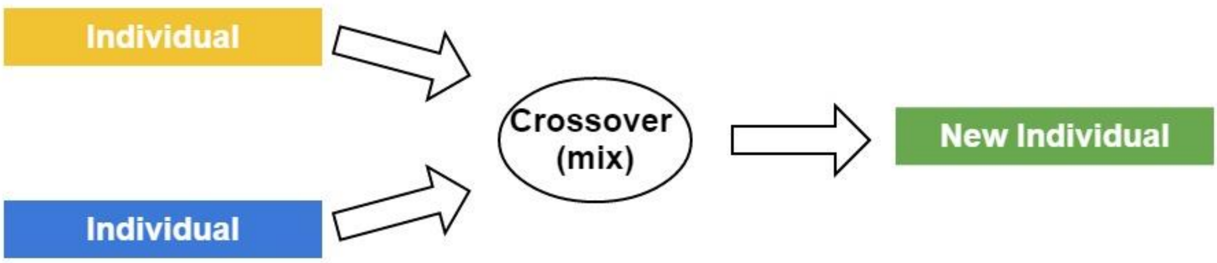


Figure 3.5: General Crossover

Mutation on the other hand requires a single candidate which can then be mutated to create new population.

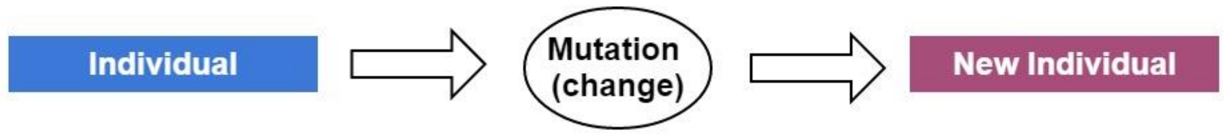


Figure 3.6: General Mutation.

As a result, the application of mutation or crossover results in the creation of new and unique individuals (Offspring). These newer individuals in the population have to compete with the older participants in the population. Based on their fitness level, the result from this competition is replicated in form of either newer individuals joining the population or being eliminated. The process reaches its conclusion when a suitable individual is identified based on the requirements (Solution) or when the program reaches the limits of its computational prowess (check section of Termination Conditions below).

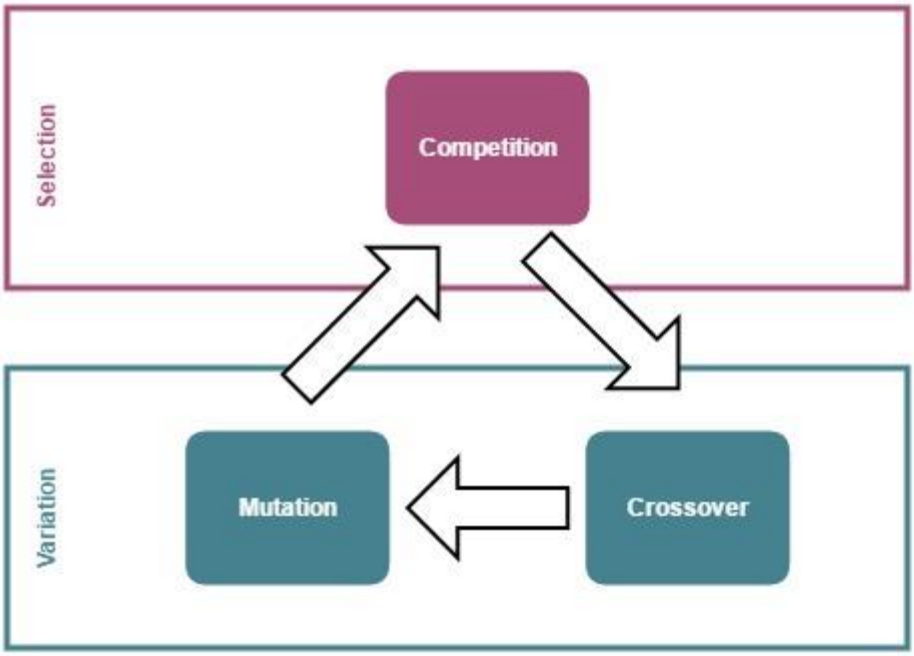


Figure 3.7: Evolutionary process.

The evolutionary process is driven by two main forces in form of variation and selection. The variation force is inculcated through mutation or crossover which propagates diversity in the population to reach an optimal solution while selection improves the quality of the mean in the population. When these two forces act simultaneously they generally result in the improvement in overall fitness of the current and subsequent populations. This entire process can be viewed as the process of evolution improving the fitness function over time in order to create the ideal or optimized solution. On the other hand, it can be also be viewed as adaptive feature where the process of evolution does not actively seek the optimization of the function but is actually driven by the environmental factors, meaning it merely expresses the prominent environmental factors. Viewing both of these perspectives, it is easy to draw the conclusion that evolution improves the viability of the population resulting in the larger and improved version of the offspring based on their environment.

To ease illustration, imagine a population of parents. Some of these parents are more capable of utilizing environment than others because they acquire preferred set of features such as strong muscles. This population of parents will mate to produce new population of offspring (children). Let's imagine that we can constrain mating process by increasing possibility of preferred parents to have more children. These preferred parents will pass their preferred features to their children. The new population of children will have more individuals which enjoy preferred features. Hence, parent population evolved to children population with incremental improvement.

It is important to realize that many factors in the evolution are heavily based on chance or have an inherent randomness in them. For example during the selection process it is not guaranteed that all the strong individuals would be selected but it is also a possibility that the weaker members of the population would also be chosen to seed the next population. These weaker individuals may have preferred few features which may greatly improve features in strong individuals if they are combined. For example, individual with strong muscles feature can utilize environment with more effectiveness if he has less important preferred feature such as short sleep period. Therefore, keeping these less important features may greatly improve future generation and lead to better final result.

Crossover process is also very random in nature; it uses a random selection criteria to decide which characteristics of the parents would continue to exist in the offspring. Same randomness also extends to the mutation where it is based on random selection to determine which parts of the parents would be modified to create a new population.

It is evident that this scheme in question is categorized in generate and test algorithms which are the algorithmic version of trial and error method. This fitness function provides an empirical or general estimate about the overall solution quality while the search process is governed by the selection and variation operators (crossover and mutation). The algorithms based on the evolution have a lot of characteristics that place them in the category of generate and test algorithms. An individual is generated based on previous population. Then, the individual is tested according to specific fitness function (minimum project lifespan). If the individual pass the test (has minimum project lifespan), it will be kept to generate new individual. This process keeps repeating to the final results. They are heavily based on the original population. Most of these algorithms use the principles of crossover where the information from two or multiple candidates is combined to create a new perspective and they are completely random or stochastic.

3.5 Components of Evolutionary Algorithms

There are numerous operators, components or procedures that must be predefined in to correctly identify an evolutionary algorithm. Some of the most important components are listed below:

- Encoding
- Fitness function
- Population
- Selection Mechanism
- Mutation and Crossover
- Termination Condition

All of the components are absolutely necessary to create a complete operable algorithm. And if we wish the algorithm to cease at some moment or point than a termination condition should also be identified.

3.5.1Encoding

In order to create an operational mechanism for resource scheduling based on evolutionary algorithm, it is important to link the real world problem with the computer world. By linking the real world problem with computer world, the algorithm creates a context or space where the evolution can find optimal solution based on the original problem. In order to bridge the difference between both worlds, it might become necessary to model the problem which can then be manipulated by the algorithm to create an optimal solution. The first problem that arises with regard to automated problem solving mechanisms is to formulate a way through which the possible solutions and be devised and stored in such a way that they can be handled by the computers. Objects that are used to create the possible solution with regard to the context of the original problem are called phenotypes while the individual participants in the evolutionary algorithm are referred as genotypes. In the context of this research, phenotypes are schedules of project tasks such as Gantt charts; while binary chromosomes are the genotypes which are a representation of schedules in form of bit strings.

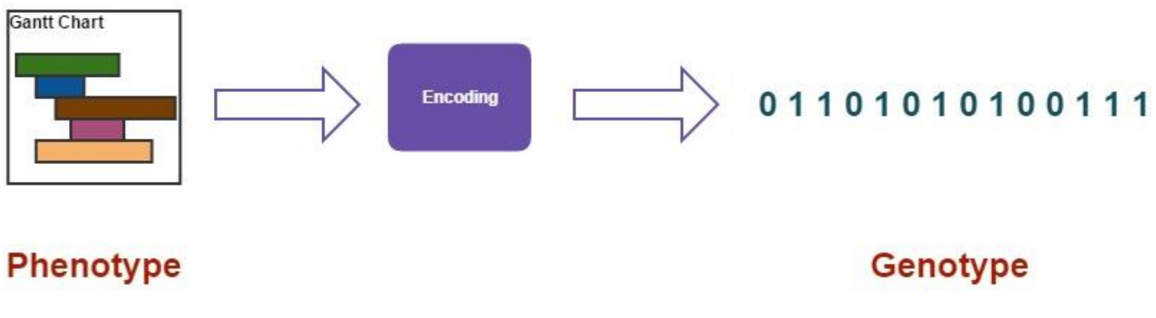


Figure 3.8: Encoding phenotype to genotype.

This first process of specifying a mapping mechanism for phenotypes onto the genotypes is called representation. It is important to note that phenotypes and genotypes differ greatly from each other, the bulk of the evolution search process occurs in Genotype space. A solution is devised by decoding the ideal genotype from the population after the termination of the program. Therefore in ideal circumstances it is desired that a solution (ideal phenotype) would be represented in the genotype space. Generally we do not have a clear idea about what an optimal solution would be therefore there should be a mechanism to represent all possible solutions. Keep in mind that we know how the fitness value of optimal solution should look like. In this research, the optimal solution (optimal scheduling decision) should have the minimum project lifespan (best fitness value). However, we are not clear on the optimal scheduling decision at beginning. We don't know how project tasks should be arranged and how resources should be distributed.

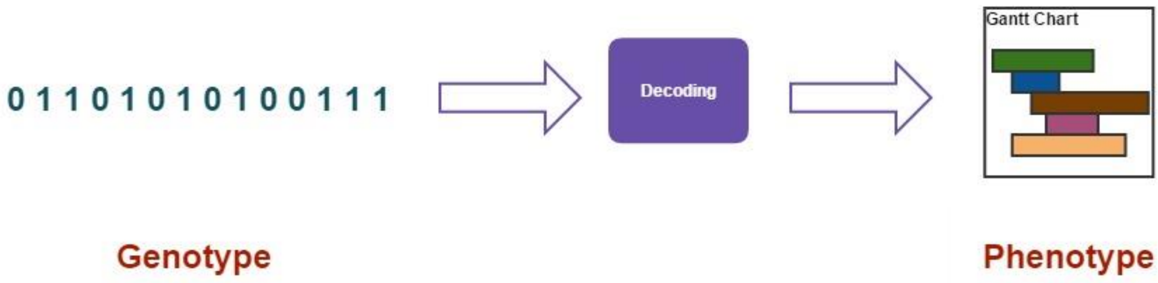


Figure 3.9: Decoding genotype to phenotype

This research relies on the application of Binary Encoding. The genotypes here are simply in form of a bit string that contains a string of binary digits. For a specific application it should be predefined about what should be the length of the string and how it can be translated to create a phenotype. In selecting a genotype and phenotype mapping mechanism for a real world problem it is important to ensure that the encoding process contains the measures to ensure that all strings create a valid solution and all the possible solutions can be represented.

Scheduling Simulation

The decoding process in this research is based event-driven simulation concept for scheduling task in project management. The proposed simulation mechanism depends on three lists. The first list contains all project tasks which are not considered for scheduling yet. This list will be called List A. The second list is called List B and it contains all project tasks which can be started since their required resources are available and their task dependency is already completed (later tasks depends on earlier ones and later tasks should not be started until earlier ones are completed). The third list is List C and it contains all project tasks which are completed after scheduling. The main purpose of simulation is to move tasks from List A to List B then to List C. Programmatically speaking, these lists are lists of computerized objects. Each object represents a task and each one of these objects has properties describing task features such as duration, resource consumption and indicators to objects of tasks that need to be completed before the task can be started.

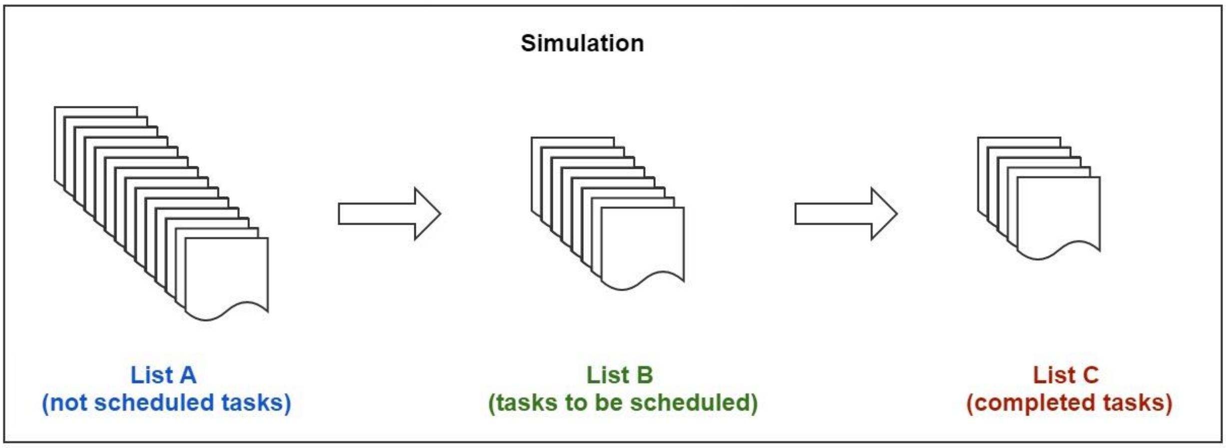


Figure 3.10: Simulation of tasks scheduling.

At the beginning of simulation, a clock will be sit to zero. This clock will be updated whenever a task is completed. This update action is called ”event”, hence event-driven simulation. Clock value will indicate the current time of simulation. Whenever an event happens (task is completed), the following steps will be performed:

1. Update resources availability after last completed task.
2. Move tasks from List A to List B if they can be started now (needed resources are available and all dependents tasks are completed).
3. Schedule tasks in List B according to Chromosome (will be explained in next section).
4. Calculate remaining time of tasks in List B by using the current value of clock.
5. Move the earliest task in List B to List C.
6. Initiate simulation event which updates the clock.
7. Repeat 1-6 until there are no tasks in List A and List B.

By the end of simulation all task in List A will end up in List C. The order of tasks in List C is controlled by chromosome value through step 3.

```
for i = 1:project.num_tasks - 2
    tasks_can_be_started = [];
    tasks_can_be_started_index = [];
    tasks_counter = 0;
    for j = 1:size(tasks_pool,2)
        cannot_be_started = 0;
        for k = 1:project.tasks(tasks_pool(j)).num_predecessors
            task_exists = 0;
            for l = 1:size(tasks_order,2)
                if project.tasks(tasks_pool(j)).predecessors(k) == tasks_order(l)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            tasks_counter = tasks_counter + 1;
            tasks_can_be_started(tasks_counter) = tasks_pool(j);
            tasks_can_be_started_index(tasks_counter) = j;
        end
    end
end
```

Figure 3.11: Select tasks from List A to List B.

```
for i = 1:project.num_tasks - 2
    gene_value = bin2dec(num2str(chromosome( (i-1)*gene_size + 1 : i*gene_size )));
    task_index = mod(gene_value, size(tasks_can_be_started,2)) + 1;
    tasks_order(i+1) = tasks_can_be_started(task_index);
    tasks_pool(tasks_can_be_started_index(task_index)) = [];
    tasks_pool = union(tasks_pool,project.tasks(tasks_order(i+1)).successors);
    for j = 1:project.tasks(tasks_order(i+1)).num_successors
        cannot_be_started = 0;
        for k = 1:project.tasks.num_predecessors
            task_exists = 0;
            for l = 1:size(tasks_order,2)
                if project.tasks.predecessors(k) == tasks_order(l)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            task_already_exists = 0;
            for k = 1:size(tasks_pool,2)
                if project.tasks(tasks_order(i+1)).successors(j) == tasks_pool(k)
                    task_already_exists = 1;
                end
            end
            if task_already_exists == 0
                tasks_pool(size(tasks_pool,2)+1) = project.tasks.successors(j);
            end
        end
    end
end
```

Figure 3.12: Schedule tasks from List B to List C.

Both of Figures 3.9 and 3.10 represent to code implementation of the proposed simulation. In Figure 3.9, the first three lines of code initiate variables which will be used as simulation memory. Then a loop procedure will be performed on every task which not scheduled. This loop tries to find tasks that can be started. Figure 3.10, shows the code that will use the output of code presented in Figure 3.9. Code in Figure 3.10 is responsible about performing the actual selection of tasks from list A. Here, genes of genetic algorithm will be used as an indication of task ranking. Hence, the first line of Figure 3.10 is to decode chromosomes genes for every task. Then, tasks will be moved from List A to List B based on their ranking as done in the main loop.

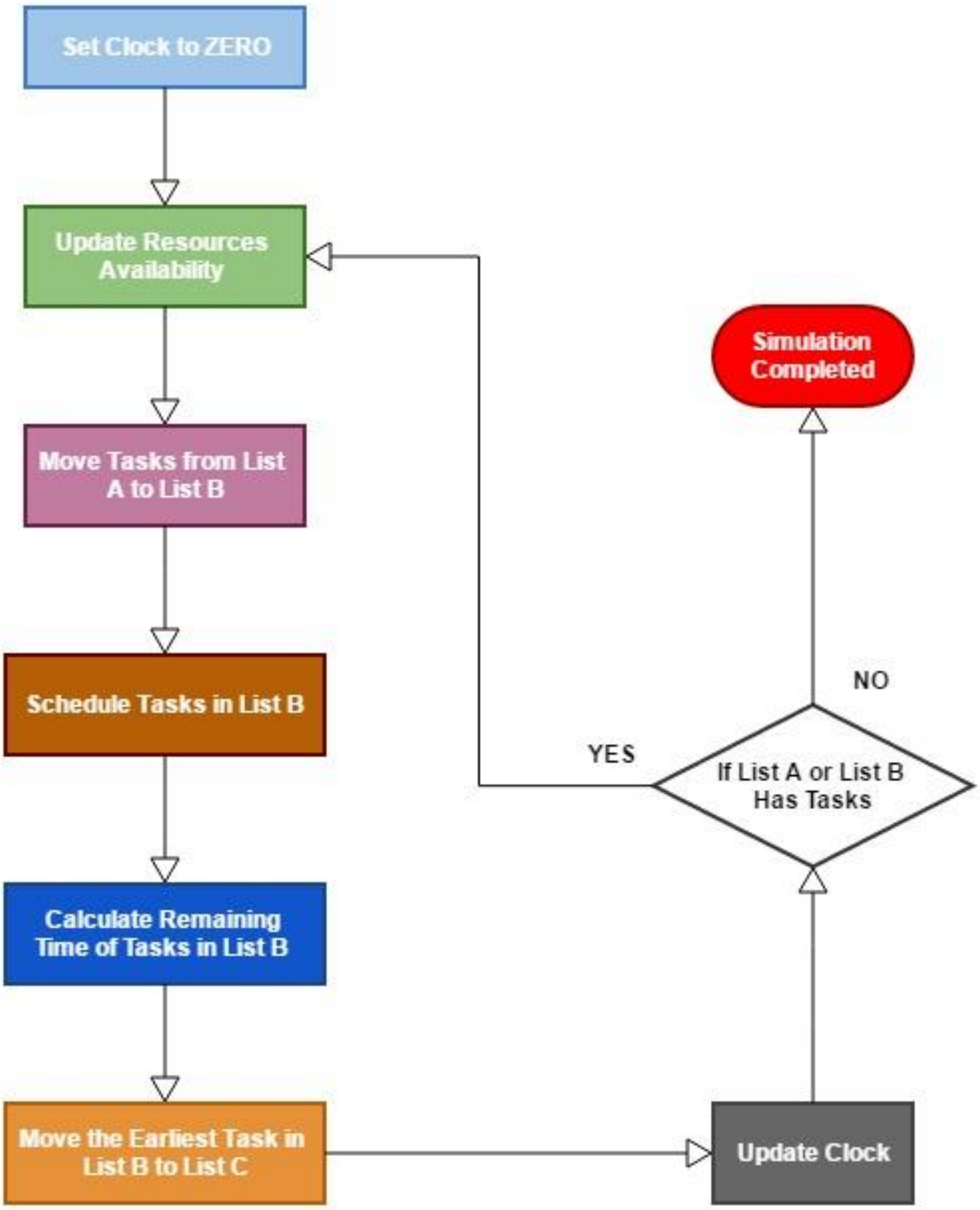


Figure 3.13: Simulation steps of tasks scheduling.

Figure 3.13, explains the proposed simulation process. At the start, the clock will be initiated. This clock will be used to track simulation time. Then status of resources availability will be update. At the start, all resources will be available. After that, code presented in Figure 3.9 and 3.10 will be executed one after the other so that tasks are moved from List A to List B. The next step will be to calculate task finish times and update the clock. The completion condition of simulation will be checked.

Task Scheduling

This research uses binary representation for chromosomes. Each chromosome is composed of genes. Number of genes depends on number of tasks in the project. For example, if there are 30 tasks in the project, there will be 30 genes in the chromosome; where each task is represented by one gene.

Figure 3.14: Chromosome of four genes.

Each gene in the chromosome is a string of binary bits which take values of zero and one only. The size of gene depends on the number of tasks in the project as well. The size of gene is the minimum number of bits needed to encode all tasks of the project. For example, a project with 30 tasks will require gene size of 5 since a gene size of 4 bits can only encode 16 tasks. Number of codes which a gene of size x can encode is 2^x. Hence, we should find an x such that $2^x \geq$ number of task in the project.

Table 3.1: Binary encoding.

x = 1	$\mathbf{2^x = 2}$
x = 2	$2^x = 4$
x = 3	$2^x = 8$
x = 4	$2^x = 16$
x = 5	$2^x = 32$

x = 6	$2^x = 64$
x = 7	$2^x = 128$
x = 8	$2^x = 256$
x = 9	$2^x = 512$
x = 10	$2^x = 1024$

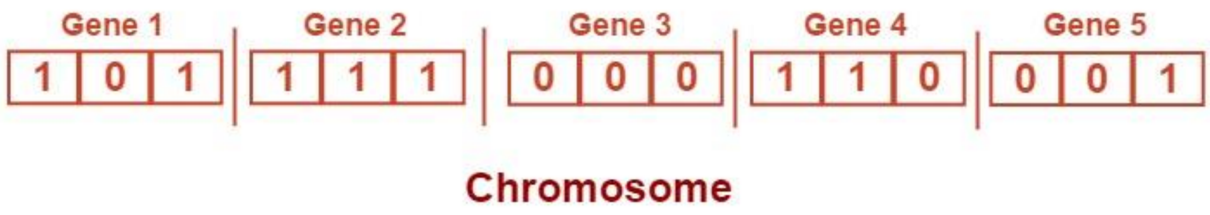


Figure 3.15: Chromosome of 5 genes with gene size of 3.

During simulation, tasks in List B will be selected based on specific gene in the chromosome. The index of chosen gene is equal to number of already scheduled tasks plus one. For example, at the beginning of simulation, the number of already scheduled tasks is 0. Hence, the index of the gene is 0 + 1 = 1. Therefore, Gene 1 will be used. If there were already 4 tasked scheduled, then Gene 5 will be used. After that values of the selected gene will be converted into number. Let gene(1) refers to the first bit in the gene; gene(2) refers to the second bit in the gene; etc. Value y will be calculated as follows:

$$y = \sum_{i=1}^{gene\ size} gene(i) \times 2^{(i-1)} + 1$$

For example, calculating y for Gene 1 in figure 10:

$$y = gene(1) \times 2^0 + gene(2) \times 2^1 + gene(3) \times 2^2 + 1 = 1 + 0 + 4 + 1 = 6$$

Value of y will be used to select the y-th task in List B for scheduling. In other words, y represents the task ranking in List B. In the previous example, y = 6, then the sixth task in List B will be scheduled next. If List B has fewer tasks than 6, modules operation will be used where y value will be updated as follows:

$$y = modules(y, size\ of\ List\ B) + 1$$

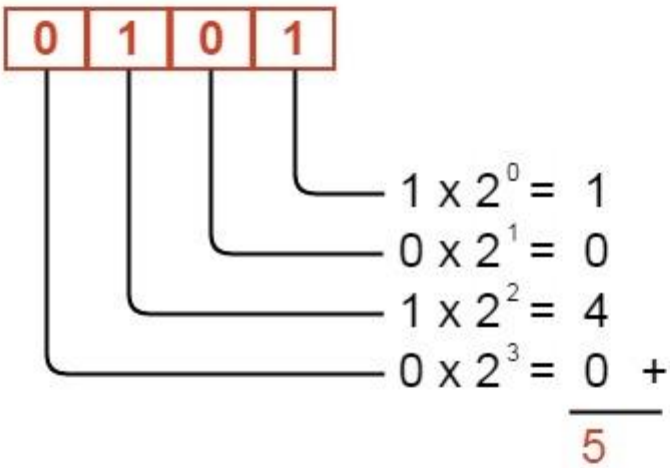


Figure 3.16: Binary to decimal.

Figure 3.14 shows an example of how binary code can be decoded into decimal number. Keep in mind that modules operation is the division reminder. The need for this operation stem from the fact that gene can be decoded into numbers that is larger than the number of available tasks. Here, using modules operation will guarantee that the gene will be decoded into number which is less than or equal the maximum available tasks.

3.5.2Fitness Function

The fitness function outlines the requirements that the members of the population must evolve or mutate to replicate. Fitness function is the very basis for the selection process and is inculcated into the algorithm in order to introduce improvement in the population. In simple terms, it dictates what signifies as improvement and sets out the goal that needs to be accomplished from the problem solving perspective. In the technical terms, it is an equation that relates quality measures for the genotypes. Usually this function is operated or designed in an inverse representation in order to create relating phenotypes. In this research minimizing project lifespan is the main fitness function in term of time. In other words, chromosomes which achieve shorter lifespan of the project will have better fitness value.

3.5.3Population

The purpose of the population is hold varying solution that might be produced through the algorithm. In simple terms a population holds multiple types of genotypes. It is basic unit where the evolution takes place. Depending on the representation, a population may be very easy to identify it might merely base on identifying the number of individuals in a given set I.e. setting the size of the population. In almost all programs based on the evolutionary algorithms the population size does not change to replicate the scarcity of the resources that in turn creates competition and forces the individual components to evolve. Selection operators or functions also mainly apply to a population and this restriction extends to survivor selection as well as parent selection. In principle these operators take account of the entire population and the resulting choices are made out of what is available in the population. In practice the best individuals are selected in order to create offspring for next population and the worst individuals are eliminated in order to improve the efficiency of the process. This selection operation runs side by side with variations functions that apply to one or multiple individuals.

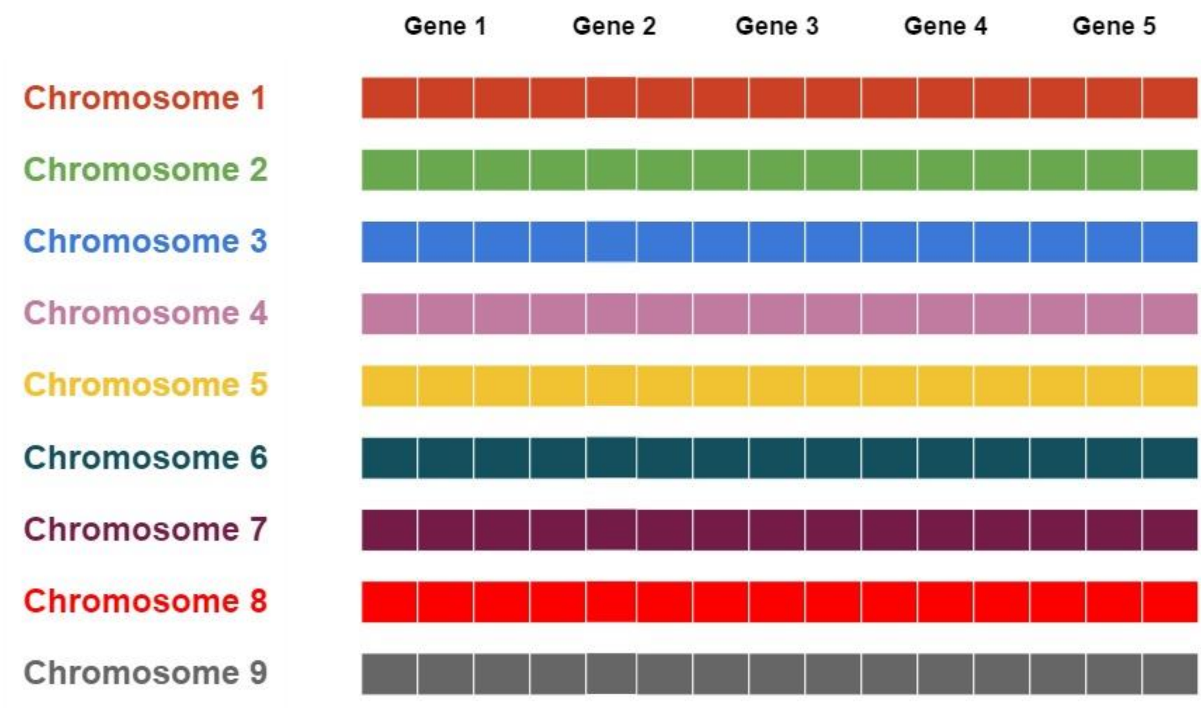


Figure 3.17: Example of population of chromosomes.

In context of project scheduling, chromosomes in figure 3.15 represents the ranking of project tasks. Each task in the project has a corresponding gene. The fist gene is for the first task and second gene is for the second task. Each one of these genes has different binary code that will be decoded to represent task ranking as decimal number.

Diversity of a population refers to the presence of multiple types of solutions within its space. There is no single concrete method of measuring diversity. Generally researchers may relate diversity with the multiple types of genotypes, fitness measures or phenotypes present within a population. Entropy is another popular measure that is widely used to measure or instill diversity in the population. In this research, diversity is insured by increasing mutation rate. It is also important to note that in certain scenarios the presence of only a single fitness value does not signify that the population lacks diversity because even that single fitness level can create many different types of phenotypes. Similar principle also applies to the presence of multiple genotypes despite a single fitness value. But contrary to that if only one genotype is present in the population it signifies that only a single phenotype along with a single fitness value exists.

3.5.4 Selection Mechanism

The purpose of the selection process is to differentiate the individuals based on their relative quality and characteristics. This function is extremely important to ensure that improved or better individuals get higher opportunity of seeding the next generation. Parent is an individual that has been subjected to the variation process in order to create an offspring. The selection process with regards to the parents is extremely important in order to facilitate improvement in quality. The parent selection process is heavily probabilistic to keep features in the population which may be useful in future generations. This ensures the individuals with the poor fitness have lower chances of getting selected as the parents while the individuals with higher fitness have higher chances of becoming parents. Even the poor fitness individuals should have small chance of being selected as parents otherwise the algorithm would become too biased and the resultant population would remain rooted in local optimum.

Based on the guidelines of Fitness Proportional Selection (FPS), the chances or probability of an individual getting selected for the parent selection process should be based on its fitness value compared to the fitness level of the entire population. In past, several problems have been identified with this selection process. Outstanding or high quality individuals can take over the entire population. This creates high level of bias in the population creating the risk that the algorithm would not conduct a thorough search in order to create solutions. This phenomenon is usually observed in populations going through their early generation and many individuals have low fitness value. This kind of biased selection is called biased convergence.

Another point to note is that when the fitness values exist very closely to each other it is highly likely that there would be little or almost no pressure on the selection process which results in a random selection. In comparison, having slightly improved fitness level is not very beneficial. Due to these reasons, it is usually observed that when a population have suffered through the convergence, the worst individuals in the population are usually eliminated and the mean of fitness of the population improves only slightly and slowly.

Ranking selection is employed in order to eliminate the inherent weaknesses of proportionate selection. It continues to exercise a perquisite selection pressure by categorizing the population based on their respective fitness level then allocating certain probability selection values to these individuals based on their overall rank. To understand this selection process let us consider a population where the individuals are ranked based on the number of worse solution existing in it, so the worst individual has the rank of 0 and the best has a rank of population size - 1. The mapping of the numbers regarding the selection probability can be achieved in various different ways including exponential decreasing or linear methods.

3.5.5 Mutation and Crossover

The aim of this operator is to facilitate the creation of newer individuals from the old ones. In the Generate and Test methods these operators serve the purpose of generation.

Mutation

This function applied to a single genotype and results in a slightly mutated child. The mutation operator is completely random and the resulting output (Child) is based upon a number of different choices that are highly random in nature. Ideally the aim of the mutation operator is to create random change that is free from any type of bias. Mutation has played different kinds of roles in different types of algorithms for examples in GA it's prime responsibility is to provide the population with fresh blood while in evolutionary programming its aim is limited to generating new individuals for the population.

Mutation has certain theoretical implications as well. It can ensure that the entire space is connected. There are theories which suggest that given time, an Evolutionary Algorithm has the capability to devise an optimal solution for any given problem. These often base themselves on the inter-connectedness of the genotype that represents a particular solution and can be devised or achieved by the application of mutation. The best way to achieve this is to give the mutation a free reign meaning that it can affect any gene.

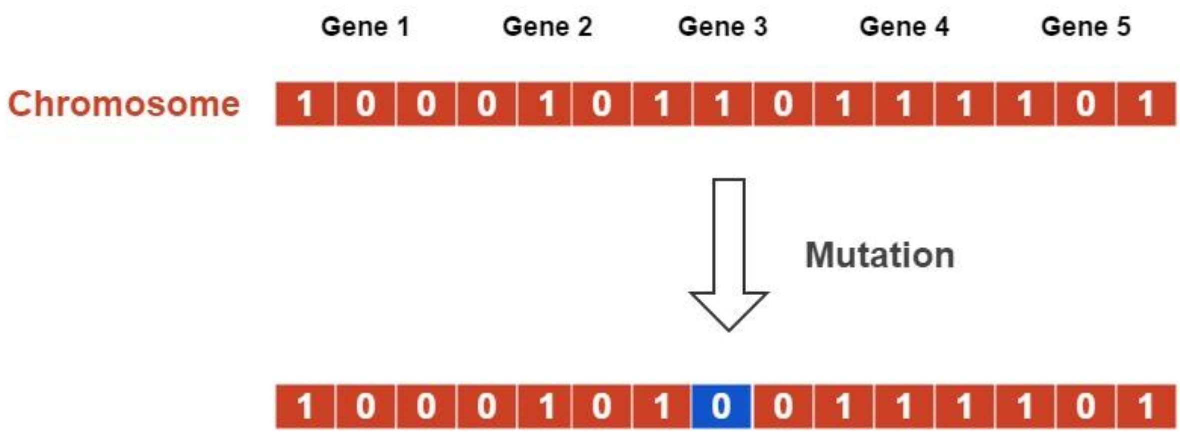


Figure 3.18: Binary mutation

There are several different mutation schemes that are used in practices but generally it is common to all. The mutation operator in binary encodings considers each bit individually and attaches a small probability of mutation to it. Therefore the chances of different values changed is not static or fixed but is determined by the sequence of number drawn randomly.

Crossover

It is evident from the name that the crossover operator uses the information from two or more genotypes and combines them together to create one or multiple unique offspring of genotypes. Similar to the mutation function the crossover operator is also completely random and the basis upon which the information is merged or created is completely random. The purpose or aim of the crossover function also greatly differs from algorithm to algorithm. In genetic programming crossover serves as the main search operator, but it is non-existent in evolutionary programming.

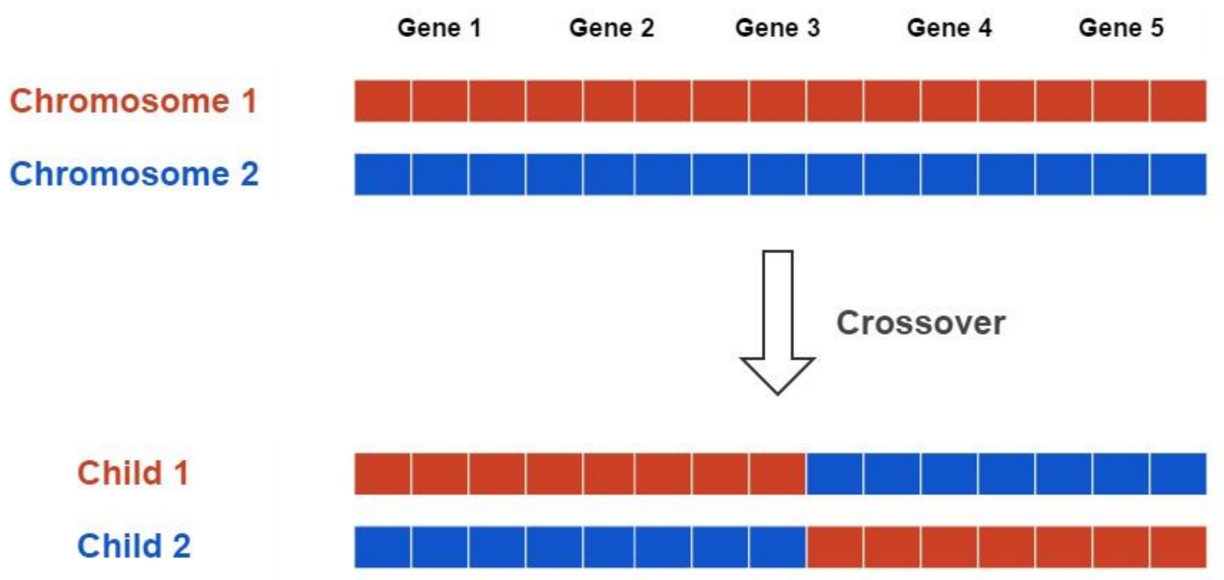


Figure 3.19: One point crossover.

The principle behind crossover is that similar to the natural selection by mating two improved individuals with another an offspring can be created that holds the improved characteristics from both of the population. Due to crossover instances may arise where the offspring is worse than the parents or it can be no better than the parents or have highly improved characteristics.

In the binary representations it is common to use 3 different kinds of crossover. All of these crossover start with at least two parents in order to create 2 sets of offspring, but there can also be examples where only a single offspring is created. Originally it was proposed that we employ **one point crossover** mechanism. It operates by choosing random ranges and then using those ranges to separate the parents and then combine them at tails to create new offspring.

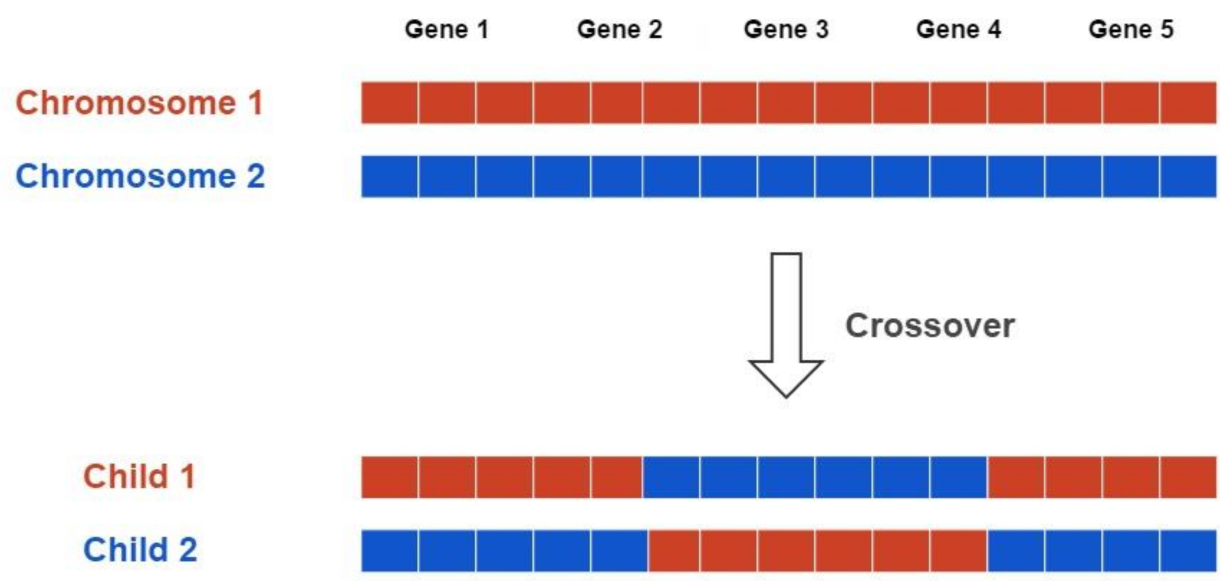


Figure 3.20: 2-point crossover.

One-point crossover can be slightly manipulated to create **N-point crossover** where the parent's chromosome are broken into multiple parts and randomly joined to create new segments. While in **uniform crossover** the function does not split the parents but rather makes a random selection about what type of characteristics the child should inherent from the offspring. This is achieved by creation of random strings and based on these strings the characteristics are inherited from one parent or the other.

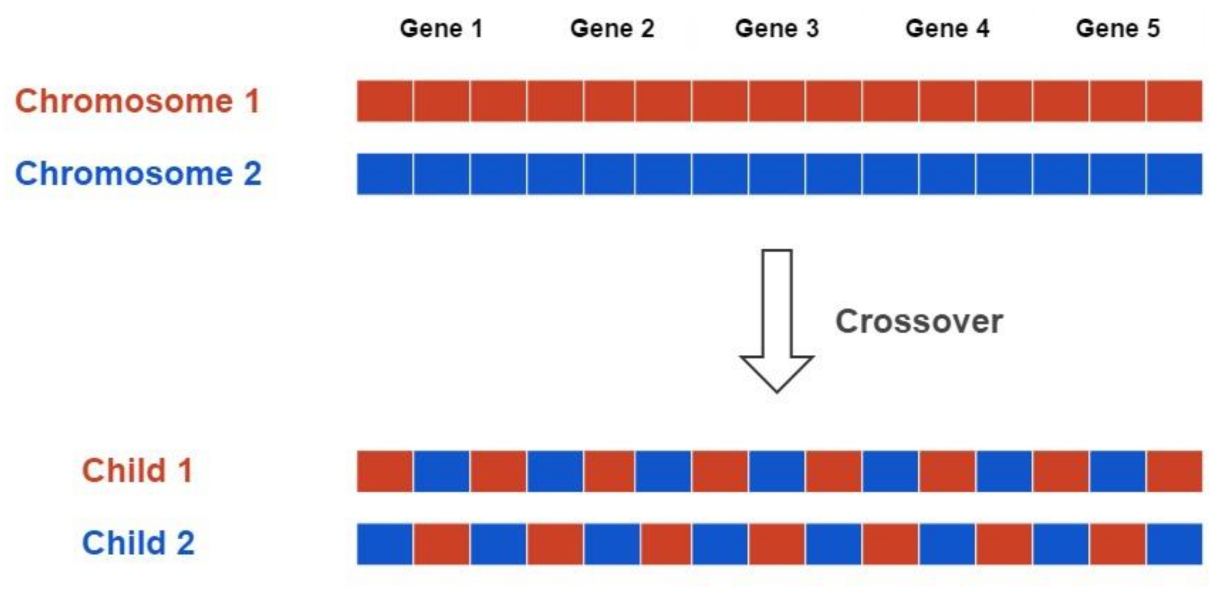


Figure 3.21: Uniform crossover.

3.5.6Termination Conditions

There are two main types of termination conditions used practically. In the most ideal circumstances a program would end when the optimum solution is reached. This usually happens when the algorithm has a known fitness level which can be achieved. In certain circumstances we may know that our model of the problem has been simplified or slightly modified in order to make it operable for the computers than the optimal solution may fall in a general range which the possibility of certain deviations and noise. But in most cases the Evolutionary algorithms have a very random nature which makes it highly unlikely to devise an optimum solution resulting in a non-ending program that never stops. Therefore it is necessary to incorporate certain conditions which eliminate or stop the program to select an acceptable solution. Some of the most common conditions used include:

- Maximum number of generations is reached.
- Program should terminate when the diversity in the population falls under a certain threshold.
- The number of fitness values reaches a certain limit.
- The improvement in the fitness level remains unchanged or under a certain threshold for a given period.
- The maximally predicted time from the CPU elapses.

In this research, a combination of termination conditions will be used. Specifically, these termination conditions are fitness limit, maximum number of generations and maximum run time. Whenever one of these conditions is satisfied, genetic algorithm will be terminated and the best available scheduling decision will used the final result.

3.6 Uncertainty

Uncertainty is the key reason behind the very need of project management. The resource allocation and project schedules are determined by estimates and these estimates are hardly single numbers ever; many assumptions openly discussed on otherwise do exist as reason behind each of these. Few of such assumptions are associated with complication involved in the tasks; others concern the ability to perform them. While few of them will lead to early completion of tasks, others will affect the execution duration by adding to it. It can be safely said at this point, that in order for a task to get completed within a decent (earliest possible time) time frame comfortably within the deadline, every favourable possibilities have to happen and every unfortunate incident anticipated need to not happen. The likelihood of this is rather low. The same situation applies on the latest date. Then best compatible date keeps up a correspondence with a scenario where in the most likely positive

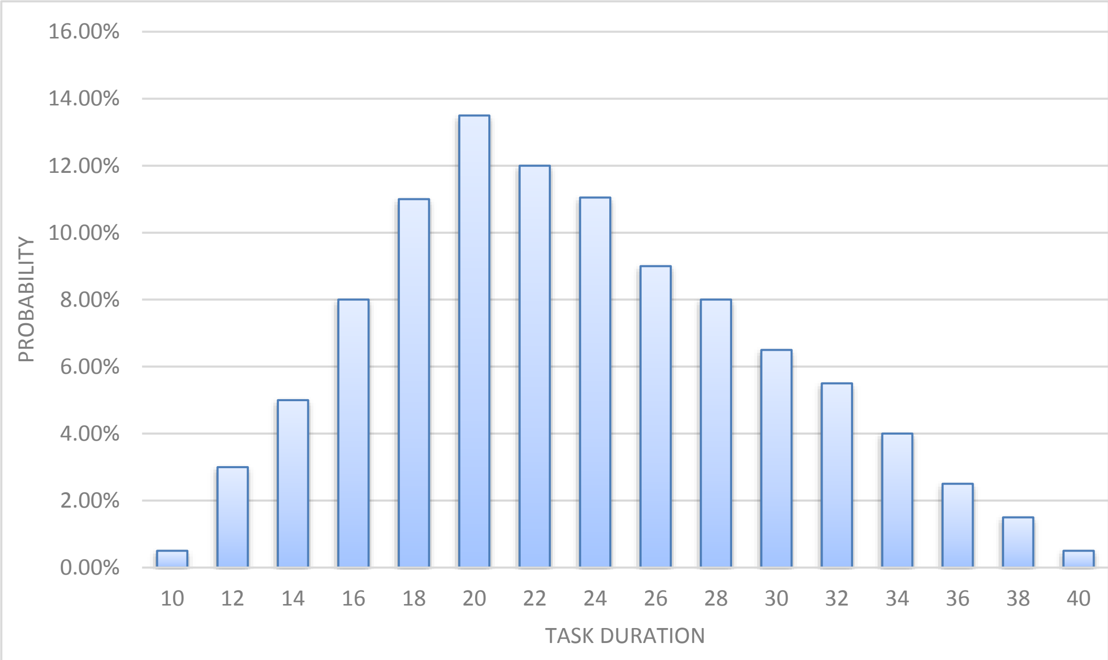
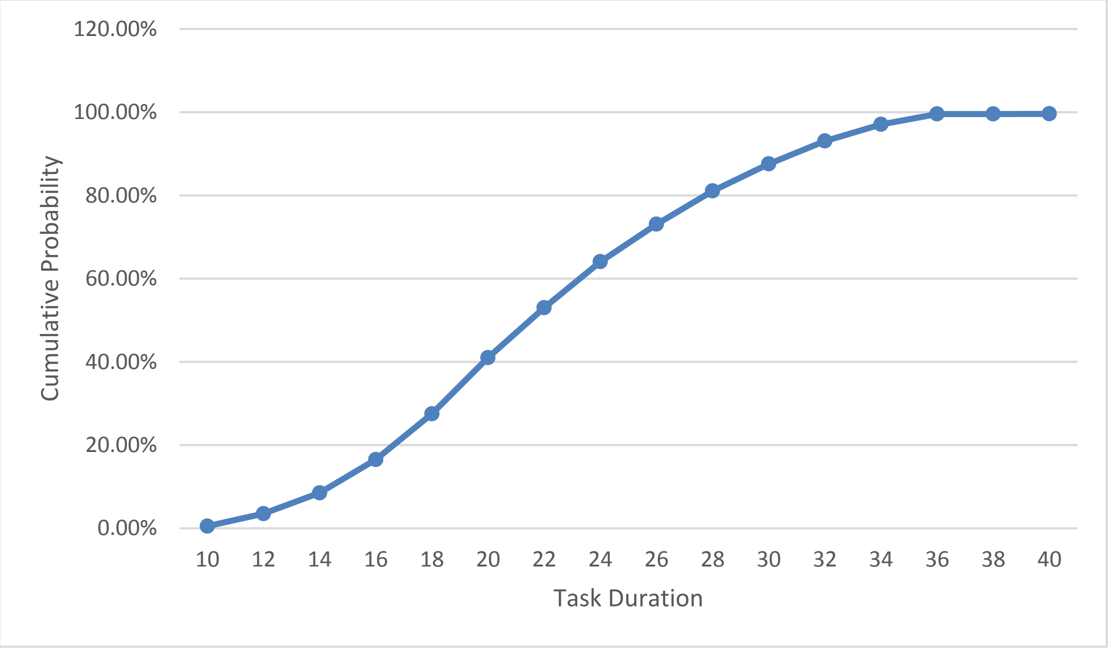


Figure 3.22: Example of task duration probability distribution.

assumptions stand true and the most like negative assumptions turn out to be false. On numerical terms, this can be explained with the help of a distribution based on triangular probability as is illustrated in the figure below.

In case every positive assumption is right and every negative assumption is wrong, the task should finish within 10 days, that being the earliest date of completion. The most realistic duration is 20 days and if every unfortunate assumption turns out to be true, then the task would take 40 days to complete, that being the latest date of completion. Given that the exact function of probability distribution for the task duration is not known, it is wise to choose a simple triangular distribution model. Also, however cynical that sounds but truth be told while there is only so much one can do to cut down on task duration, the possibility of things going wrong is endless. If one looks at the crisis from the perspective of project management, the importance of probability of completing a task on a particular date is far less than probability of completing the task within or on a certain date. It is this probability of task completion that is called the on-time probability and it is possible to derive it from the cumulative distribution illustrated in the following figure.



Possibilities based on cumulative calculations. The almost probable date of completion is characterised by an on-time probability which is below 40%. The estimated date of completion is approximately 23 days. In case one needs to be assured as high as 75% that the task till end within time, then the time schedules has to be 27 days and no more. Usually, the number of estimations determining duration of task is directly proportional to the time span between earliest and latest possible dates of completion. This kind of uncertainty leads to on-time probabilities of highly different nature.

In case of shifting from tasks to projects, central limit theorem is highly resorted to by conventional critical path planning in order to have access to the probable uncertainty associated with the project. This theorem states, the allocation of the total number of many independent and erratic variables arrives at a normal method of distribution with variables increasing in numbers. By using this approach to closely examine project uncertainly, during calculation one has to consider the estimate project time duration as the addition of the estimated duration of task coupled with critical path, there being a deviation that is standard and equivalent to the square root of the addition of the squares of the standard deviation that characterises the same tasks. At this point a normal distribution can also be resorted to in order to find out the on-time probability for the given project.

Uncertainty is incorporated through simulation. Imagine that we have a task with 100 unit of time as a duration. If we would like to incorporate 5 % uncertainty, then the task duration will be possible in the interval of 95-105 time units. The same task can have 10 different durations because of uncertainty. We are looking for scheduler that can schedule such task. Hence, GA is used to find the best scheduler for all possible duration values. The same level of uncertainty will be applied to all tasks in the project. Therefore, for any deterministic project, we can generate countless numbers of the same project structure but with different task durations.

3.7 Proposed Optimizer

Based on discussions in previous sections, the proposed solution starts with random population of chromosomes. Each one of these chromosomes is decoded into scheduling decision for the project under consideration. Then, the lifespan of the project based on each generated scheduling decision is calculated. Values of lifespan will be used as the fitness values for corresponding chromosomes which generated the scheduling decisions. The best set of chromosomes which achieved the minimum lifespan will be selected. These selected chromosomes will generate new population through crossover and mutation. After that the same steps will be repeated on the new population. Thought out this process, the best chromosomes so far is always saved as the algorithm solution.

ex

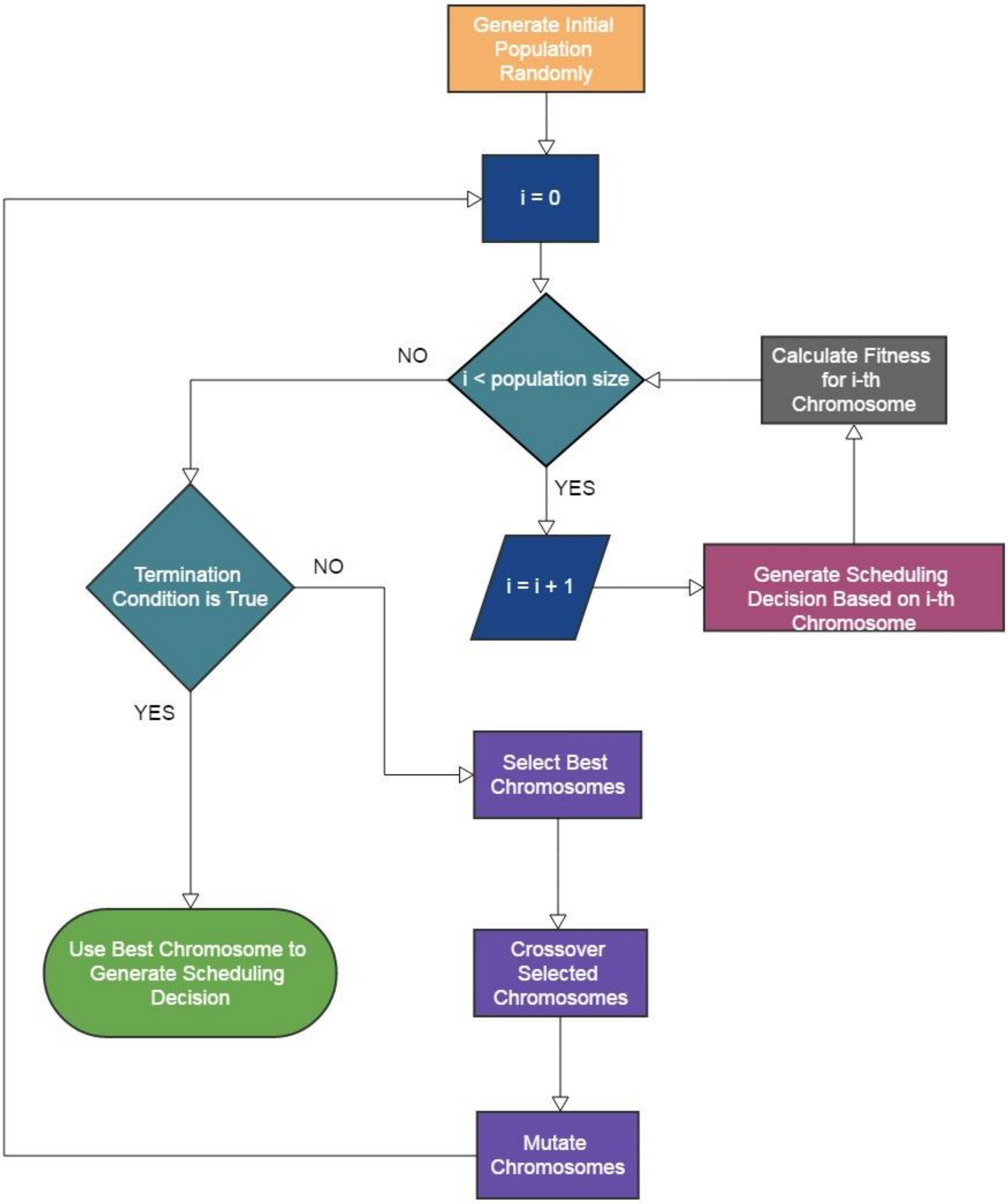


Figure 3.23: Flow chart of proposed mechanism.

Keep in mind that resources scheduling are handled in simulation when tasks are selected from List A to be put in List B. A task cannot be put in List B unless its all need resources are available. If the resources are not available, the task will be kept in List A. In addition, genetic algorithm can be instructed to optimize resources usage rather than project lifespan. This can be done by modifying fitness function.

3.8 Chapter Summary

This chapter proposed a scheduling solution for resources scheduling problem in project management. The proposed solution is designed to handle the uncertainty aspect of project management. A modified implementation of binary genetic algorithm was proposed. In addition, a simulator to perform resource scheduling was developed.

Chapter 4: Results

4.1 Introduction

This chapter presents an extensive evaluation and testing for the proposed resources scheduling algorithm in this thesis. At the beginning, a broad discussion regarding simulation input is provided. During this discussion, several aspects will be covered such as data distribution and general statistics. In addition, discussions regarding projects networks and structure are provided. Later on, simulation output will be discussed where all necessary statistical analysis is performed. Furthermore, resources consumption during simulation is discussed as well. This chapter concludes by providing direct comparison of the proposed solution with existing counterparts in literature.

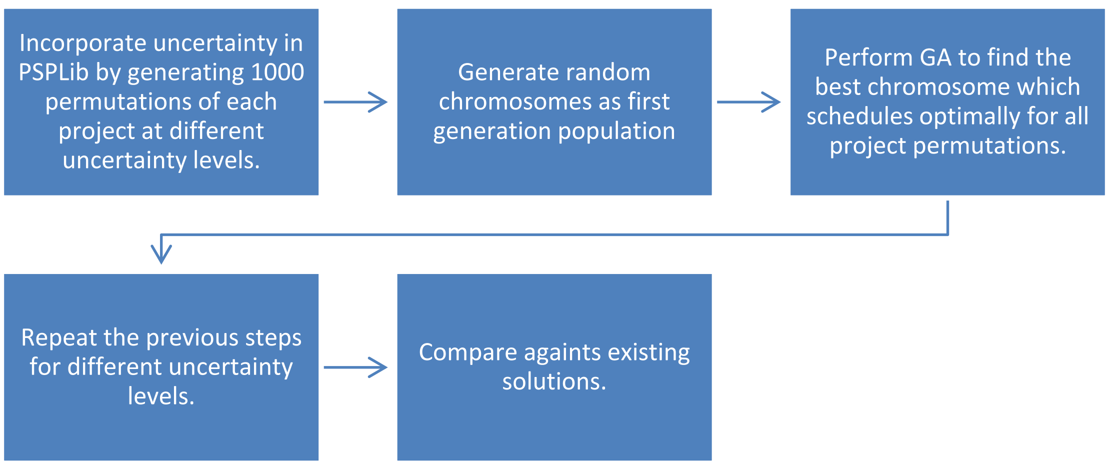


Figure 4.1: Adopted analysis approach.

The proposed scheduling mechanism in this thesis is evaluated based in two aspects. First, the theoretical and technical was investigated. Second, evidence of the accuracy and applicability of the proposed mechanisms is established. The results are reported in three phases. The first phase handled the development and the analysis of the proposed mechanisms (see chapter 3). The second phase handled numerical evaluations and testing. This phase is called Simulation and Testing phase. Finally, the third phase investigated the applicability and the adoption of the proposed mechanisms by normal managers in real projects management environment. As a result, the third phase required a real case study to be completed. This phase is called Applicability Evaluation phase.

4.2 Simulation and Testing Phase

In this phase, the proposed mechanisms is evaluated using simulation approach. The reason behind using this approach of testing is to investigate the real performance of the proposed mechanisms without the effects of other factors which are found in reality such as manager preferences. Sometimes good resources scheduling mechanisms can be developed which have very good performance on paper. However, managers in the real world do not adopt these mechanisms because they do not understand them or they cannot incorporate them in their management style. This issue (adoption in real work place) will be addressed in the next phase. Nevertheless, the performance of the proposed resources scheduling mechanisms has to be proven before they can be introduced in real project portfolio management environment. This is the sole purpose of this phase.

4.2.1 Implementation and Coding

In this activity, an actual implementation of the proposed resources scheduling mechanisms using Matlab environment is performed. Matlab is chosen because it is a very powerful developing environment for solution based on mathematical programming techniques. Also, it has very good support for artificial intelligence techniques as well. The support for mathematical programming and artificial intelligence techniques is provided in the form of toolboxes. By using these toolboxes, the researcher can reduce the coding time greatly. Also, Matlab provides toolboxes for the statistical analysis which is the preferred approach for managers in the real world since they have a good background in statistical theory. Every aspect of the proposed resources scheduling mechanisms can be traced and evaluated in very efficient way.

4.2.2Data Set and Simulation Environment

The simulation experiments are conducted using PSPLIB (Kolisch et al, 1996) which is a well-known benchmark in resources scheduling literature. This benchmark includes set of projects where the each project has a set of tasks. The resource dependency for each task in the benchmark projects is predefined and fixed. Therefore, different resources scheduling mechanisms can be compared by equating their performances when they are used on the same set of projects in the benchmark. Using this benchmark allow us to compare the proposed resources scheduling mechanisms with the previously published solutions in the literature.

A high-performance computation environment is utilised using cloud computation approach. The reason behind using cloud computation is the fact that the adopted search technique, which is genetic algorithm, requires a huge amount of computation resources and a lot of execution time. Acquiring the necessary hardware to perform simulation experiments would be very expensive. A cheaper and more effective approach is to use cloud services. Many scientists in many fields around the globe to perform their simulation experiments have used these services. Amazon Web Services (AWS) was used as the main provider of cloud services. AWS provider servers with many sizes and computation power. The server used in this research is EC2 C4.8xlarge which has 36 processing units that can be used to perform distributed computing.

To achieve optimum performance from the proposed scheduler, it is critical to devise the parameter setting with regards to the approach or methodology outlined in chapter 3 because different parameters and the population are the crucial elements for optimum performance.

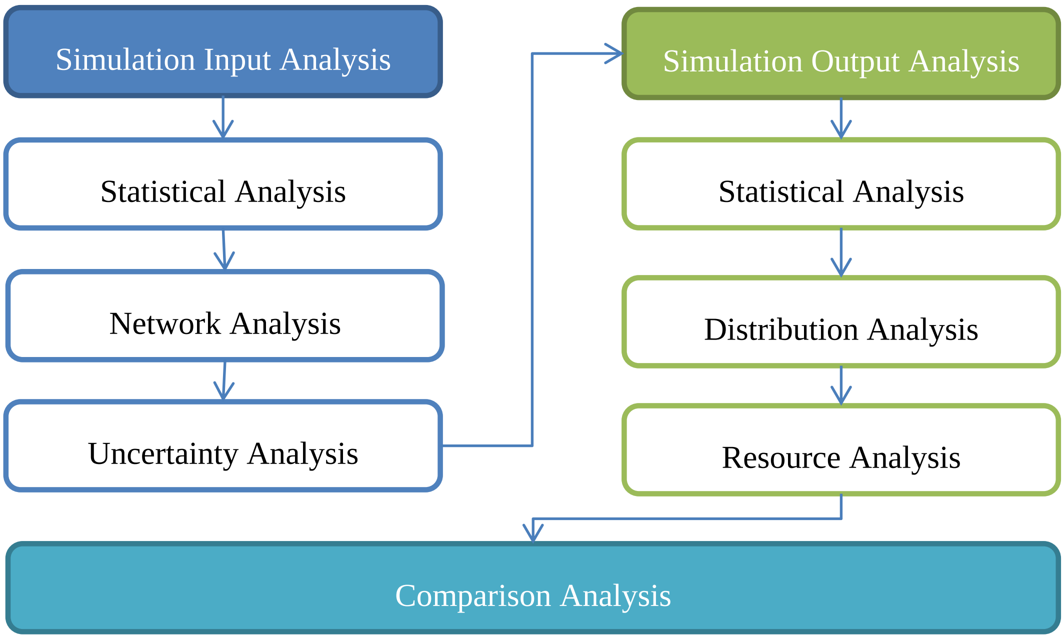


Figure 4.2: Stage of data analysis process.

4.3 **Input Data for Simulation**

The research used three projects from PSPLib database as input for simulation. Each one of these projects has different sizes. The configuration of these projects is illustrated in the following graph.

These statistics provide insights regarding complexity of each of the projects. For example, when the average of task duration is very close to the maximum value of this duration, one can conclude that task in this project has homogenous duration. In other words, most task in the project have similar durations. The same logic can be applied to all statistics of all project characteristics to extract more information regarding projects under consideration.

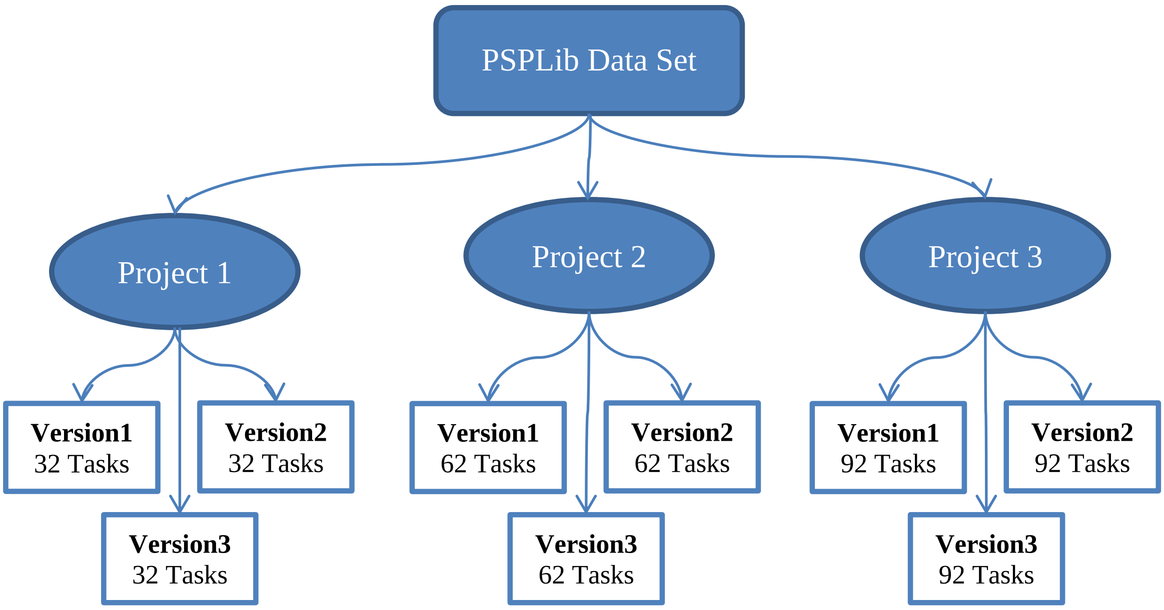


Figure 4.3: Illustrative diagram of used dataset.

The reason behind choosing three projects with three versions is to cover the entire complexity spectrum as possible as it can be. The first project is the smallest in term of tasks number while the third project is the largest. On the other hand, the first version of each project is the simplest in term of task dependency on each other while the third is the most complex.

4.3.1Project One

The first project has 32 tasks. It also has three versions where each version has unique structure and resources dependency. The characteristics of the first project are shown in Table A.1. This table shows the duration for each task in addition to each task dependency with regard to resources consumption.

Table 4.1: Resources dependency of first version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	4	0	0	0
3	4	10	0	0	0
4	6	0	0	0	3
5	3	3	0	0	0
6	8	0	0	0	8
7	5	4	0	0	0
8	9	0	1	0	0
9	2	6	0	0	0
10	7	0	0	0	1
11	9	0	5	0	0
12	2	0	7	0	0
13	6	4	0	0	0
14	3	0	8	0	0
15	9	3	0	0	0
16	10	0	0	0	5
17	6	0	0	0	8
18	5	0	0	0	7
19	3	0	1	0	0
20	7	0	10	0	0
21	2	0	0	0	6
22	7	2	0	0	0
23	2	3	0	0	0
24	3	0	9	0	0
25	3	4	0	0	0
26	7	0	0	4	0
27	8	0	0	0	7
28	3	0	8	0	0
29	7	0	7	0	0
30	2	0	7	0	0
31	2	0	0	2	0
32	0	0	0	0	0

Table 4.2: Statistics for version 1 of project 1.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.266667	1.433333	2.1	0.2	1.5
Standard Error	0.503891	0.41787	0.601551	0.137939	0.48539

Median	5	0	0	0	0
Mode	3	0	0	0	0
Standard Deviation	2.850439	2.363832	3.402887	0.780302	2.745781
Sample Variance	8.125	5.587702	11.57964	0.608871	7.539315
Kurtosis	-1.20652	4.684914	0.07056	20.14368	1.130044
Skewness	0.040462	2.045148	1.348826	4.430277	1.668811
Range	10	10	10	4	8
Minimum	0	0	0	0	0
Maximum	10	10	10	4	8
Sum	158	43	63	6	45
Count	32	32	32	32	32

The first analysed project data is task's duration. The average of duration is 5.27. The standard deviation of duration is 2.58. All tasks in this version of this project have a minimum of duration at 2. The maximum of duration is 10. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of duration is 15. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of duration is 15.

The second analyzed project data is task's resources consumption. There are four resources types. Resource 1 is analyzed first. The average of consumption for first resource is 1.43. The standard deviation of consumption for first resource is 2.38. The minimum of consumption for first resource is 0. All tasks in this version of this project have a maximum of consumption for first resource at 10. The number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 10. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 20.

Resource 2 is analyzed second. All tasks in this version of this project have an average of consumption for second resource at 2.1. The standard deviation of consumption for second resource is 3.42. Analysis shows that the minimum of consumption for second resource is 0. The maximum of consumption for second resource is 10. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for second resource at 8. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for second resource at 22.

Resource 3 is analyzed third. Investigation of data leads to the fact that the average of consumption for third resource is 0.2. All tasks in this version of this project have a standard deviation of consumption for third resource at 0.79. A further investigation of data leads to the fact that the minimum of consumption for third resource is 0. Analysis shows that the maximum of consumption for third resource is 4. The number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 2. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 28.

Resource 4 is analyzed fourth. The average of consumption for fourth resource is 1.5. All tasks in this version of this project have a standard deviation of consumption for fourth resource at 2.77. Analysis shows that the minimum of consumption for fourth resource is 0. It also shows that the maximum of consumption for fourth resource is 8. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource at 7. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 23.

The next step is to analyze project structure. There are two parameters which can be used to measure project complexity. These two parameters are Number of Successors for each task and Number of Predecessors. Structure for first version of first project is shown in Figure 4.3.

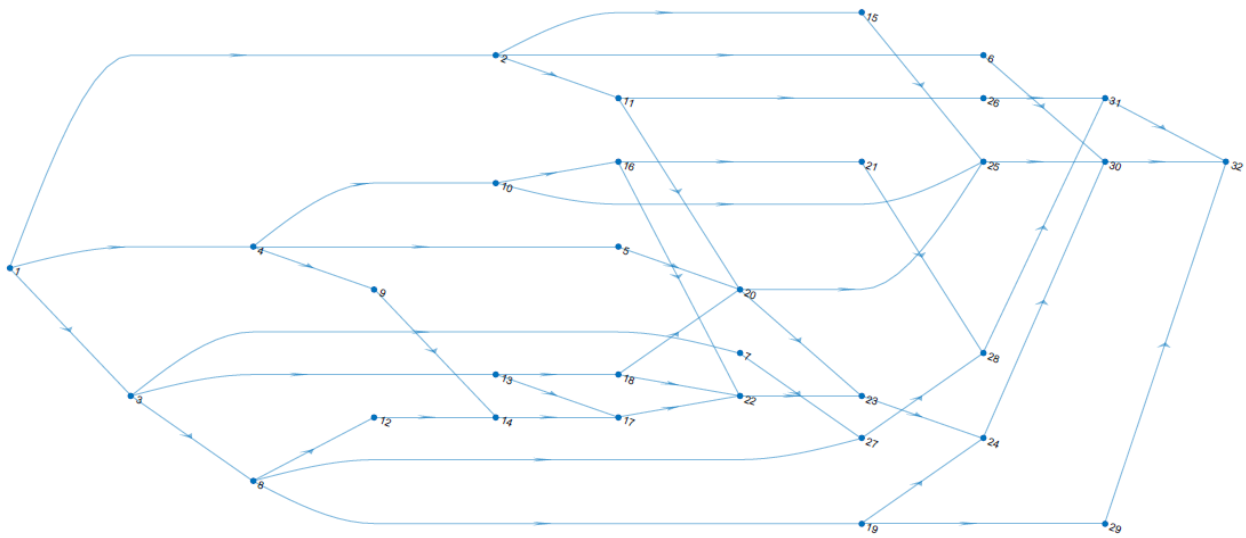


Figure 4.4: Structure for first version of first project.

Number of Successors is analyzed 4. The average of number of successors is 1.5. A further investigation of data leads to the fact that the standard deviation of number of successors is 0.72. Analysis shows that the minimum of number of successors is 1.0. In addition, it shows that the maximum of number of successors is 3.0. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of number of successors at 11. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of successors is 19.

Now, Number of Predecessors will be analyzed. Analysis shows that the average of number of predecessors is 1.5. The standard deviation of number of predecessors is 0.72. A further investigation of data leads to the fact that the minimum of number of predecessors is 1.0. It also leads to the fact that the maximum of number of predecessors is 3.0. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of number of predecessors is 11. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of predecessors at 19.

The characteristics of the second project are shown in Table A.2.

Table 44.3: Resources dependency of second version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	2	3	3	8
3	5	2	3	0	1
4	9	7	0	0	0
5	3	5	1	0	1
6	6	4	8	1	0
7	8	0	10	0	0
8	8	0	0	0	5
9	10	5	0	0	0
10	1	0	7	0	0
11	5	2	3	0	0
12	7	10	6	0	8
13	10	0	0	4	0
14	3	1	0	6	0
15	2	7	7	0	0
16	6	3	10	0	4
17	4	2	10	0	5
18	5	10	0	3	0
19	6	8	0	9	4
20	6	1	0	0	0
21	7	4	7	0	0
22	3	0	6	0	9
23	8	3	0	3	5
24	8	0	0	7	0
25	4	3	8	0	0
26	2	2	1	0	0
27	3	0	0	0	2
28	5	1	7	9	0
29	3	0	0	9	7
30	2	0	5	0	8
31	8	0	0	0	2
32	0	0	0	0	0

This table shows the duration for each task in addition to each task dependency with regard to resources consumption. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of duration at 15. The number of tasks in the lower 50th percentile with reference to average of duration is 15.

Table 4.4: Statistics for version 2 of project 1.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.15625	2.5625	3.1875	1.6875	2.15625
Standard Error	0.498201	0.534943	0.654447	0.532109	0.540887
Median	5	2	1	0	0
Mode	8	0	0	0	0
Standard Deviation	2.818251	3.026096	3.702114	3.010064	3.059721
Sample Variance	7.94254	9.157258	13.70565	9.060484	9.361895
Kurtosis	-0.91907	0.601198	-1.2065	1.425496	-0.25152
Skewness	-0.08691	1.21596	0.635277	1.66071	1.118138
Range	10	10	10	9	9
Minimum	0	0	0	0	0
Maximum	10	10	10	9	9
Sum	165	82	102	54	69
Count	32	32	32	32	32

All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for first resource at 12. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 18. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of consumption for second resource is 12. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for second resource at 18. The number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 9. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 21. The number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource is 10. The number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 20.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for second version of first project is shown in Figure 4.4.

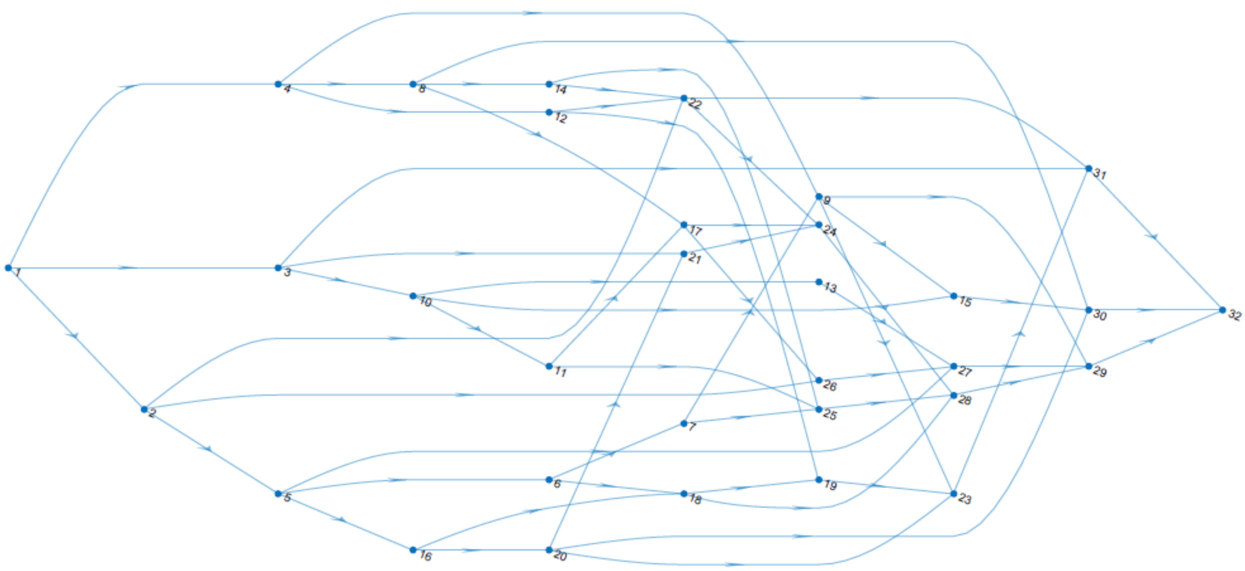


Figure 4.5: Structure for second version of first project.

Number of Successors is analyzed 4. All tasks in this version of this project have an average of number of successors at 1.83. A further investigation of data leads to the fact that the standard deviation of number of successors is 0.82. All tasks in this version of this project have a minimum of number of successors at 1.0. Analysis shows that the maximum of number of successors is 3.0. The number of tasks in the upper 50th percentile with reference to average of number of successors is 17. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of successors is 13. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of number of predecessors at 16. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of predecessors is 14. All tasks in this version of this project have a maximum of duration at 10. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of duration is 17. The number of tasks in the lower 50th percentile with reference to average of duration is 13.

Table 4.5: Resources dependency of third version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	7	3	10	4	4
3	10	5	8	9	8
4	2	4	4	3	9
5	6	7	5	9	10
6	8	4	7	3	4
7	1	2	6	9	7
8	8	3	7	9	8
9	1	10	4	8	3
10	7	10	10	3	1
11	4	4	1	4	5
12	6	2	6	6	2
13	4	7	7	8	10
14	9	7	9	8	8
15	10	8	1	3	9
16	1	4	6	2	7
17	7	10	2	8	5
18	8	6	3	3	8
19	6	7	9	1	5
20	2	6	2	1	2
21	5	6	5	9	1
22	6	2	3	9	7
23	2	2	10	8	4
24	2	10	10	7	7
25	6	6	7	9	5
26	9	8	6	4	7
27	4	7	5	7	7

28	10	2	10	9	4
29	3	7	9	6	6
30	5	10	4	6	4
31	8	9	4	4	9
32	0	0	0	0	0

Table 4.6: Statistics for version 3 of project 1.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.21875	5.5625	5.625	5.59375	5.5
Standard Error	0.54944	0.534943	0.5513	0.538553	0.511954
Median	6	6	6	6	5.5
Mode	6	7	10	9	7
Standard Deviation	3.108099	3.026096	3.118623	3.046514	2.896048
Sample Variance	9.660282	9.157258	9.725806	9.28125	8.387097
Kurtosis	-1.15634	-0.97679	-0.9737	-1.25921	-0.82458
Skewness	-0.13507	-0.11763	-0.16463	-0.36775	-0.33999
Range	10	10	10	9	10
Minimum	0	0	0	0	0
Maximum	10	10	10	9	10
Sum	167	178	180	179	176
Count	32	32	32	32	32

A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 18. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 12.

All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for second resource at 13. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for third resource at 18. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 12. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource at 16. The number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 14.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for third version of first project is shown in Figure 4.5.

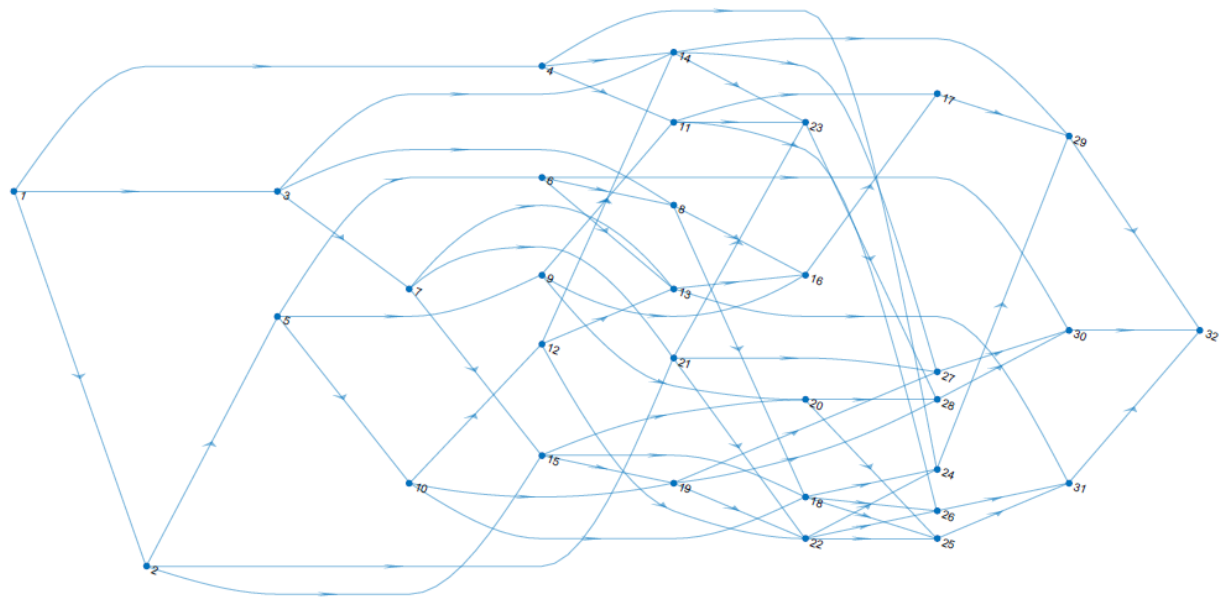


Figure 4.6: Structure for third version of first project.

Number of Successors is analyzed 4. All tasks in this version of this project have an average of number of successors at 2.17. The standard deviation of number of successors is 0.93. A further investigation of data leads to the fact that the minimum of number of successors is 1.0. The maximum of number of successors is 3.0. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of number of successors at 16. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of successors is 14.

Now, Number of Predecessors will be analyzed. Analysis shows that the average of number of predecessors is 2.17. A further investigation of data leads to the fact that the standard deviation of number of predecessors is 0.86. The minimum of number of predecessors is 1.0. All tasks in this version of this project have a maximum of number of predecessors at 3.0. The number of tasks in the upper 50th percentile with reference to average of number of predecessors is 14. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of predecessors is 16.

4.3.2Project Two

The second project has 62 tasks. It also has three versions where each version has unique structure and resources dependency. The first version is represented in Table A.4. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of duration is 31. The number of tasks in the lower 50th percentile with reference to average of duration is 29.

Table 4.7: Statistics for version 1 of project 2.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.306452	1.387097	1.290323	1.032258	1.241935
Standard Error	0.409814	0.367474	0.319792	0.293566	0.336942
Median	5.5	0	0	0	0
Mode	3	0	0	0	0
Standard Deviation	3.226878	2.893491	2.518047	2.311537	2.653087
Sample Variance	10.41274	8.37229	6.340561	5.343205	7.038868
Kurtosis	-1.37964	2.551264	1.786721	3.761471	2.938239
Skewness	0.047018	1.96587	1.779965	2.230746	2.08133
Range	10	10	9	9	9
Minimum	0	0	0	0	0

Maximum	10	10	9	9	9
Sum	329	86	80	64	77
Count	62	62	62	62	62

The number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 14. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 46. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for second resource at 14. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for second resource is 46. The number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 11. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 49. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource at 11. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 49.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for first version of second project is shown in Figure 4.6.

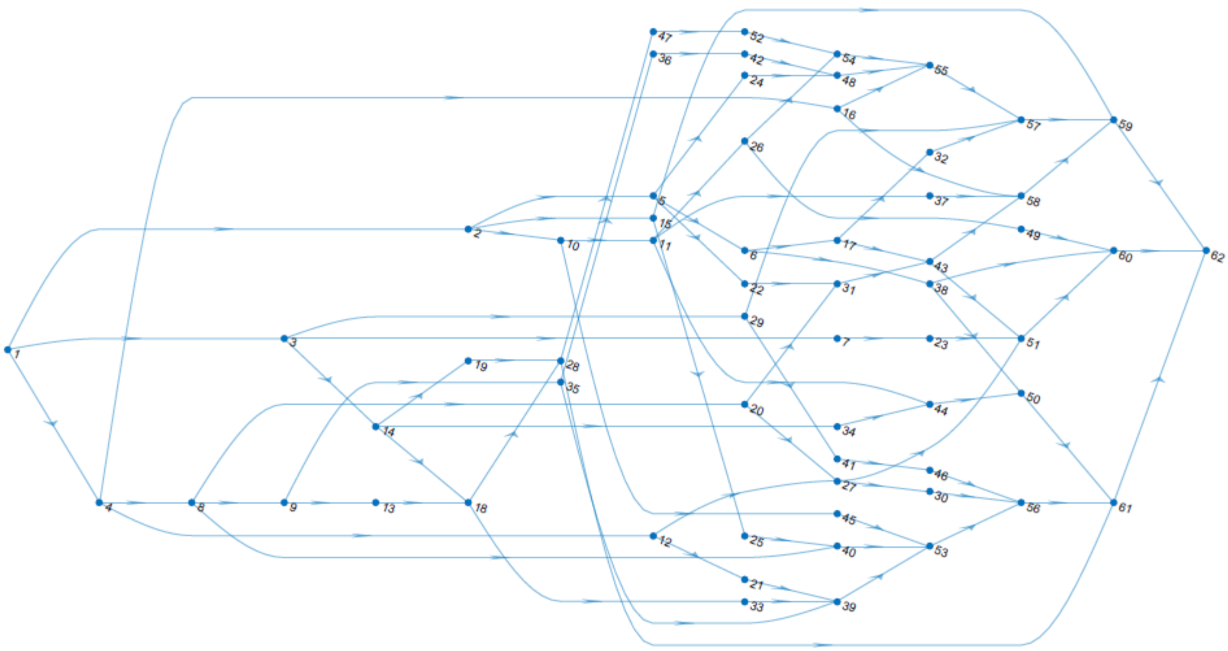


Figure 4.7: Structure for first version of second project.

Number of Successors is analyzed 4. Analysis shows that the average of number of successors is 1.5. Analysis shows that the standard deviation of number of successors is 0.7. All tasks in this version of this project have a minimum of number of successors at 1.0. The maximum of number of successors is 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of successors is 23. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of successors at 37.

Now, Number of Predecessors will be analyzed. A further investigation of data leads to the fact that the average of number of predecessors is 1.5. The standard deviation of number of predecessors is 0.76. A further investigation of data leads to the fact that the minimum of number of predecessors is 1.0. All tasks in this version of this project have a maximum of number of predecessors at 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of predecessors is 20. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of predecessors is 40.

The characteristics second version of project 2 are shown in Table A.5. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of duration is 28. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of duration is 32.

Table 4.8: Statistics for version 2 of project 2.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.096774	2.741935	2.548387	2.951613	2.483871
Standard Error	0.374144	0.400248	0.451613	0.441398	0.411843
Median	5	1.5	0	1.5	0
Mode	8	0	0	0	0
Standard Deviation	2.946015	3.151557	3.556004	3.47557	3.242858
Sample Variance	8.679006	9.932311	12.64516	12.07959	10.51613
Kurtosis	-1.13549	-0.83395	-0.32139	-0.96143	-0.68752
Skewness	0.029311	0.739309	1.090532	0.724655	0.893029
Range	10	10	10	10	10
Minimum	0	0	0	0	0
Maximum	10	10	10	10	10
Sum	316	170	158	183	154
Count	62	62	62	62	62

All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for first resource at 30. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 30. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for second resource at 22. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for second resource is 38. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for third resource at 25. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 35. The number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource is 23. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 37.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for second version of second project is shown in Figure 4.7.

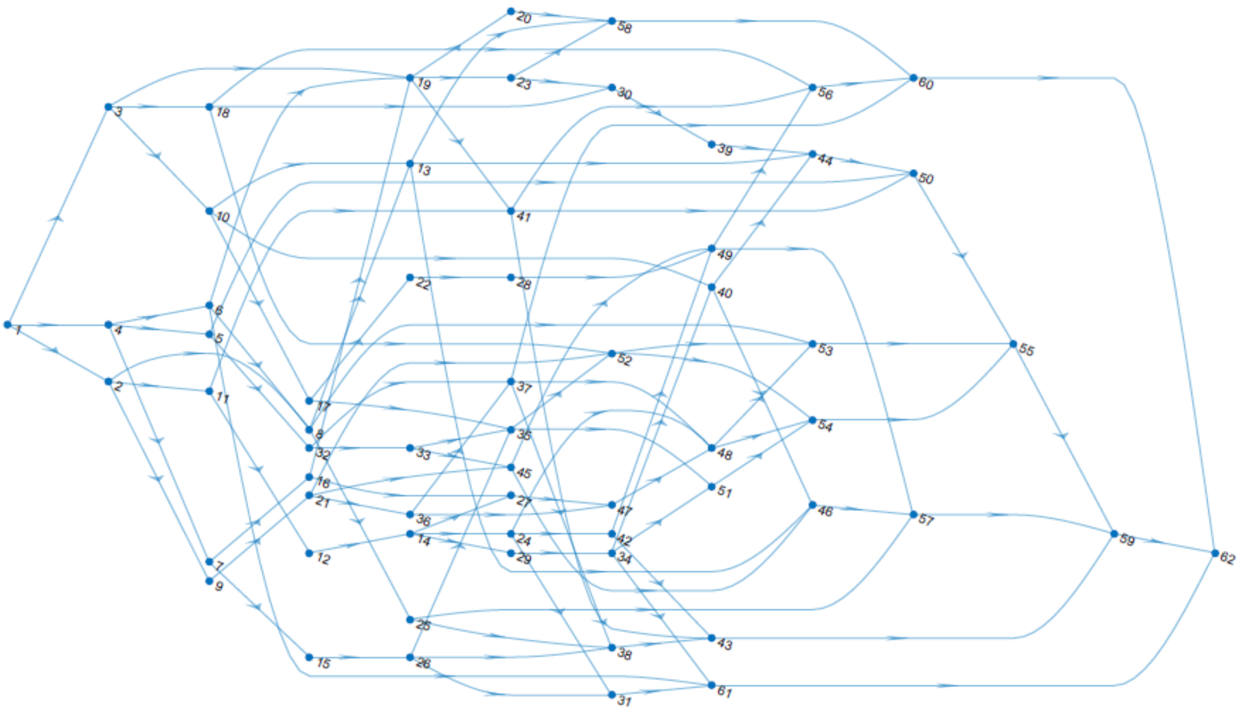


Figure 4.8: Structure for second version of second project.

Number of Successors is analyzed 4. All tasks in this version of this project have an average of number of successors at 1.82. The standard deviation of number of successors is 0.83. All tasks in this version of this project have a minimum of number of successors at 1.0. A further investigation of data leads to the fact that the maximum of number of successors is 3.0. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of number of successors is 33. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of successors at 27.

Now, Number of Predecessors will be analyzed. Analysis shows that the average of number of predecessors is 1.82. All tasks in this version of this project have a standard deviation of number of predecessors at 0.88. Analysis shows that the minimum of number of predecessors is 1.0. Analysis shows that the maximum of number of predecessors is 3.0. The number of tasks in the upper 50th percentile with reference to average of number of predecessors is 30. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of predecessors at 30.

The characteristics of third version of project 2 are presented in Table A.6. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of duration is 24. The number of tasks in the lower 50th percentile with reference to average of duration is 36.

Table 4.9: Statistics for version 3 of project 2.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.016129	5.516129	5	5.129032	5.387097
Standard Error	0.389234	0.414404	0.382731	0.403242	0.393868
Median	4	5.5	5	5	5.5
Mode	4	10	3	1	9

Standard Deviation	3.064829	3.263016	3.01363	3.175128	3.101321
Sample Variance	9.393178	10.64728	9.081967	10.08144	9.618191
Kurtosis	-1.17178	-1.29824	-1.12135	-1.32549	-1.19788
Skewness	0.347921	-0.05177	0.237669	0.065524	-0.06527
Range	10	10	10	10	10
Minimum	0	0	0	0	0
Maximum	10	10	10	10	10
Sum	311	342	310	318	334
Count	62	62	62	62	62

The number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 31. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 29. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of consumption for second resource is 25. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for second resource is 35. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 28. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for third resource at 32. The number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource is 31. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource at 29.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for third version of second project is shown in Figure 4.8.

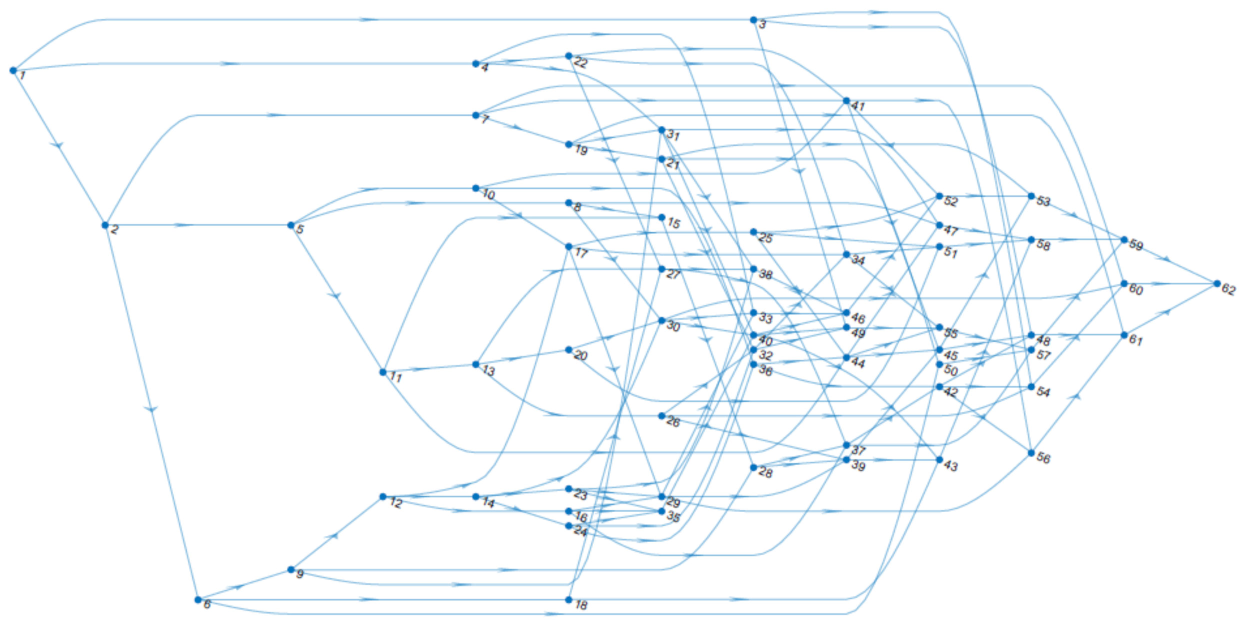


Figure 4.9: Structure for third version of second project.

Number of Successors is analyzed 4. A further investigation of data leads to the fact that the average of number of successors is 2.13. The standard deviation of number of successors is 0.9. A further investigation of data leads to the fact that the minimum of number of successors is 1.0. The maximum of number of successors is 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of successors is 29. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of number of successors is 31.

Now, Number of Predecessors will be analyzed. All tasks in this version of this project have an average of number of predecessors at 2.13. A further investigation of data leads to the fact that the standard deviation of number of predecessors is 0.94. Analysis shows that the minimum of number of predecessors is 1.0. The maximum of number of predecessors is 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of predecessors is 31. The number of tasks in the lower 50th percentile with reference to average of number of predecessors is 29.

4.3.3Project Three

The third project has 92 tasks. It also has three versions where each version has unique structure and resources dependency. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of duration is 49. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of duration is 41.

Table 4.10: Statistics for version 1 of project 3.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.51087	1.021739	1.565217	1.73913	1.206522
Standard Error	0.31047	0.247287	0.301812	0.330728	0.256561
Median	6	0	0	0	0
Mode	8	0	0	0	0
Standard Deviation	2.977921	2.371897	2.894877	3.172234	2.46085
Sample Variance	8.868012	5.625896	8.380315	10.06307	6.055781
Kurtosis	-1.30813	3.761295	1.203993	0.699311	3.562608
Skewness	-0.17443	2.261628	1.626302	1.504792	2.085549
Range	10	9	10	10	10
Minimum	0	0	0	0	0
Maximum	10	9	10	10	10
Sum	507	94	144	160	111
Count	92	92	92	92	92

Analysis shows that the number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 17. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for first resource at 73. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of consumption for second resource is 24. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for second resource at 66. The number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 23. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for third resource at 67. The number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource is 21. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 69.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for first version of third project is shown in Figure 4.9.

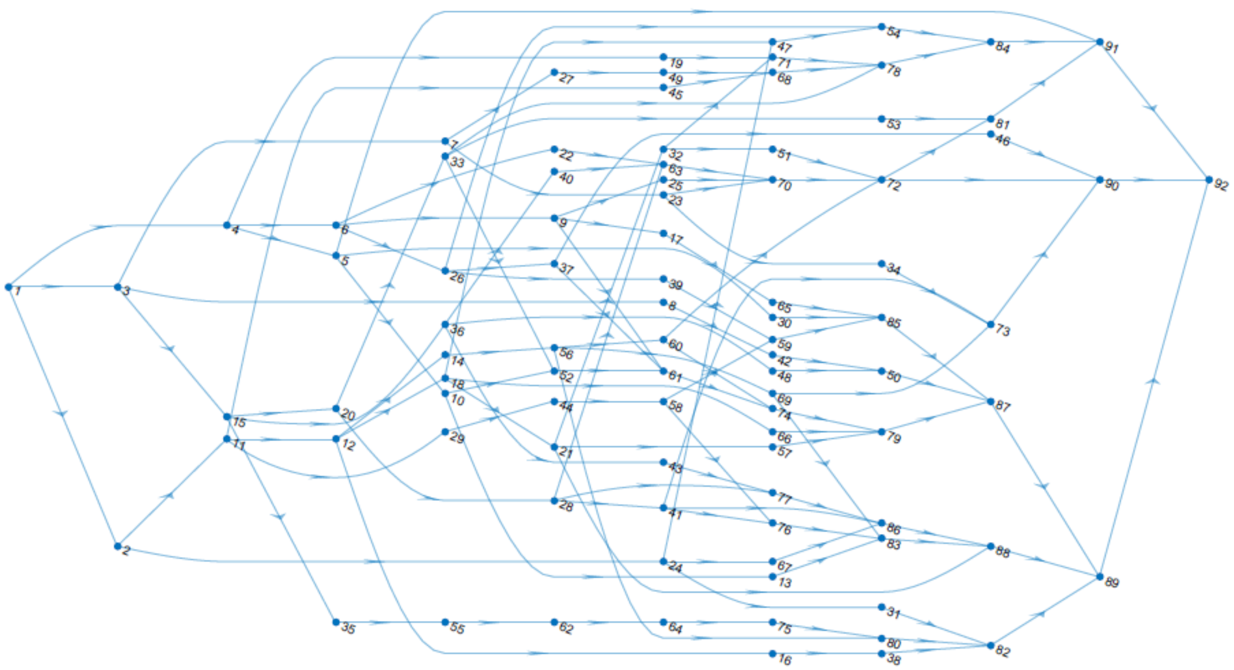


Figure 4.10: Structure for first version of third project.

Number of Successors is analyzed 4. Analysis shows that the average of number of successors is 1.5. All tasks in this version of this project have a standard deviation of number of successors at 0.79. The minimum of number of successors is 1.0. All tasks in this version of this project have a maximum of number of successors at 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of successors is 28. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of number of successors is 62.

Now, Number of Predecessors will be analyzed. The average of number of predecessors is 1.5. A further investigation of data leads to the fact that the standard deviation of number of predecessors is 0.78. The minimum of number of predecessors is 1.0. All tasks in this version of this project have a maximum of number of predecessors at 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of predecessors is 29. The number of tasks in the lower 50th percentile with reference to average of number of predecessors is 61.

The second version is represented in Table A.8. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of duration is 45. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of duration is 45.

Table 4.11: Statistics for version 2 of project 3.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.163043	2.728261	3.445652	2.521739	2.652174
Standard Error	0.310386	0.343217	0.394166	0.360985	0.351923
Median	5	0	1.5	0	0
Mode	8	0	0	0	0

Standard Deviation	2.977119	3.292026	3.780703	3.462446	3.375525
Sample Variance	8.863235	10.83743	14.29372	11.98853	11.39417
Kurtosis	-1.3078	-0.90619	-1.36643	-0.50688	-0.68777
Skewness	-0.03657	0.744581	0.503421	1.018426	0.894836
Range	10	10	10	10	10
Minimum	0	0	0	0	0
Maximum	10	10	10	10	10
Sum	475	251	317	232	244
Count	92	92	92	92	92

The number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 39. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for first resource at 51. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for second resource at 43. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for second resource is 47. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for third resource at 33. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 57. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource is 36. The number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 54.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for second version of third project is shown in Figure 4.10.

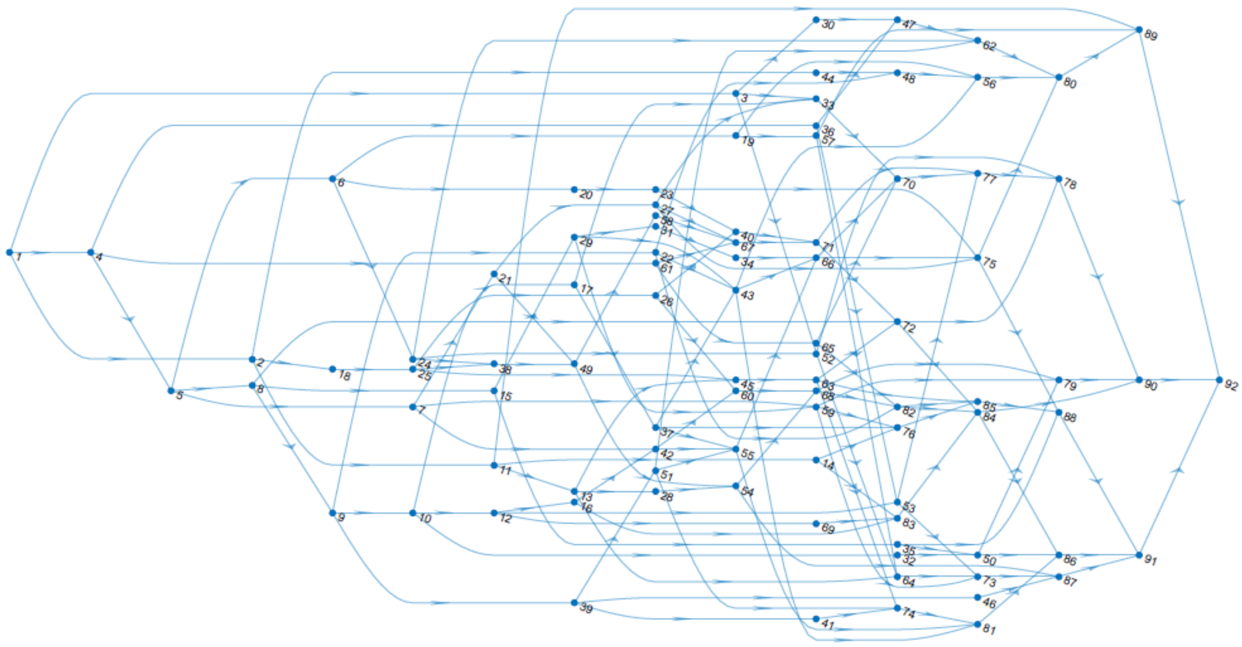


Figure 4.11: Structure for second version of third project.

Number of Successors is analyzed 4. A further investigation of data leads to the fact that the average of number of successors is 1.81. All tasks in this version of this project have a standard deviation of number of successors at 0.82. A further investigation of data leads to the fact that the minimum of number of successors is 1.0. All tasks in this version of this project have a maximum of number of successors at 3.0. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of number of successors is 50. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of successors at 40.

Now, Number of Predecessors will be analyzed. Analysis shows that the average of number of predecessors is 1.81. All tasks in this version of this project have a standard deviation of number of predecessors at 0.89. Analysis shows that the minimum of number of predecessors is 1.0. The maximum of number of predecessors is 3.0. A further investigation of data leads to the fact that the number of tasks in the upper 50th percentile with reference to average of number of predecessors is 44. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of number of predecessors is 46.

The third version is represented in Table A.9. This table shows the duration for each task in addition to each task dependency with regard to resources consumption. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of duration at 45. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of duration is 45.

Table 4.12: Statistics for version 3 of project 3.

Statistics	Duration	Resource 1	Resource 2	Resource 3	Resource 4
Mean	5.51087	5.467391	4.869565	5.445652	5.336957
Standard Error	0.319943	0.32253	0.314188	0.323607	0.304166
Median	5	6	4.5	5.5	5.5
Mode	4	3	1	8	9
Standard Deviation	3.068789	3.093599	3.013586	3.103929	2.917462
Sample Variance	9.417463	9.570354	9.081701	9.634376	8.511586
Kurtosis	-1.33539	-1.29658	-1.31223	-1.4674	-1.22188
Skewness	0.018387	-0.03625	0.167293	-0.04706	-0.17076
Range	10	10	10	10	10
Minimum	0	0	0	0	0
Maximum	10	10	10	10	10
Sum	507	503	448	501	491
Count	92	92	92	92	92

Analysis shows that the number of tasks in the upper 50th percentile with reference to average of consumption for first resource is 47. A further investigation of data leads to the fact that the number of tasks in the lower 50th percentile with reference to average of consumption for first resource is 43. The number of tasks in the upper 50th percentile with reference to average of consumption for second resource is 46. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of consumption for second resource at 44. Analysis shows that the number of tasks in the upper 50th percentile with reference to average of consumption for third resource is 46. The number of tasks in the lower 50th percentile with reference to average of consumption for third resource is 44. All tasks in this version of this project have a number of tasks in the upper 50th percentile with reference to average of consumption for fourth resource at 46. Analysis shows that the number of tasks in the lower 50th percentile with reference to average of consumption for fourth resource is 44.

The next step is to analyze project structure. As said before, there are two parameters which can be used to measure project complexity which are Number of Successors for each task and Number of Predecessors. Structure for third version of third project is shown in Figure 4.11.

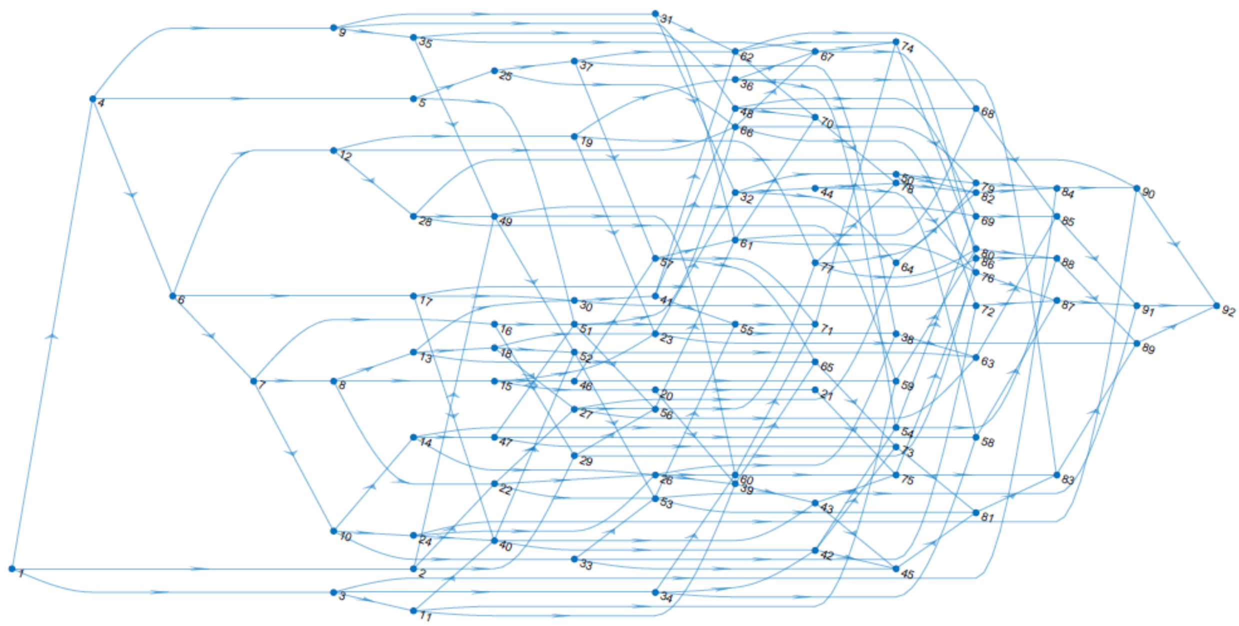


Figure 4.12: Structure for third version of third project.

Number of Successors is analyzed 4. All tasks in this version of this project have an average of number of successors at 2.12. Analysis shows that the standard deviation of number of successors is 0.89. A further investigation of data leads to the fact that the minimum of number of successors is 1.0. All tasks in this version of this project have a maximum of number of successors at 3.0. The number of tasks in the upper 50th percentile with reference to average of number of successors is 42. The number of tasks in the lower 50th percentile with reference to average of number of successors is 48.

Now, Number of Predecessors will be analyzed. Analysis shows that the average of number of predecessors is 2.12. All tasks in this version of this project have a standard deviation of number of predecessors at 0.93. The minimum of number of predecessors is 1.0. The maximum of number of predecessors is 3.0. The number of tasks in the upper 50th percentile with reference to average of number of predecessors is 45. All tasks in this version of this project have a number of tasks in the lower 50th percentile with reference to average of number of predecessors at 45.

4.4 Uncertainty Analysis

The uncertainty analysis is conducted by assuming that task duration is following Weibull distribution. This distribution is chosen because it has the ability to assume different aspects of other distributions. For example, if the data actually follows normal distribution, Weibull distribution will model this data to high level of accuracy (Krishnamoorthy, 2016). This distribution requires two main parameters which are scale and shape parameters. In addition, rate of consumption with regard to each resource depending on this distribution can be very helpful in uncertainty analysis since it help us to establish a direct link between the availability of resource and tasks durations. This analysis is conducted for each project in the data set where averaging is performed over project versions. The uncertainty analysis for first project is provided in Table A.10 in appendix.

Analysis shows that the minimum duration mean is 3.0. The maximum duration mean is 8.33. A further investigation of data leads to the fact that the minimum standard deviation of duration is 0.58. Analysis shows that the maximum standard deviation of duration is 4.93. The minimum scale parameter is 3.41. All tasks in this project have a maximum scale parameter at 8.57. A further investigation of data leads to the fact that the minimum shape parameter is 1.08. Analysis shows that the maximum shape parameter is 23.91.

Analysis shows that the average for consumption rate of first resource type is 16.95 as calculated from data shown in Table A.10. All tasks in this project have a standard deviation for consumption rate of first resource type at 9.37. A further investigation of data leads to the fact that the minimum for consumption rate of first resource type is 6.0. The maximum for consumption rate of first resource type is 42.0.

All tasks in this project have an average for consumption rate of second resource type at 19.19. Analysis shows that the standard deviation for consumption rate of second resource type is 10.83. The minimum for consumption rate of second resource type is 5.78. A further investigation of data leads to the fact that the maximum for consumption rate of second resource type is 50.0.

All tasks in this project have an average for consumption rate of third resource type at 13.29. Analysis shows that the standard deviation for consumption rate of third resource type is 7.9. All tasks in this project have a minimum for consumption rate of third resource type at 1.67. A further investigation of data leads to the fact that the maximum for consumption rate of third resource type is 36.0.

All tasks in this project have an average for consumption rate of fourth resource type at 16.78. A further investigation of data leads to the fact that the standard deviation for consumption rate of fourth resource type is 10.01. Analysis shows that the minimum for consumption rate of fourth resource type is 3.33. The maximum for consumption rate of fourth resource type is 36.11.

The uncertainty analysis for second project is provided in Table A.11. A further investigation of data leads to the fact that the minimum duration mean is 2.33. All tasks in this project have a maximum duration mean at 8.67. Analysis shows that the minimum standard deviation of duration is 0.58. All tasks in this project have a maximum standard deviation of duration at 5.2. All tasks in this project have a minimum scale parameter at 2.63. The maximum scale parameter is 9.4. Analysis shows that the minimum shape parameter is 1.14. Analysis shows that the maximum shape parameter is 27.11.

The average for consumption rate of first resource type is 16.88. A further investigation of data leads to the fact that the standard deviation for consumption rate of first resource type is 10.49. Analysis shows that the minimum for consumption rate of first resource type is 1.89. The maximum for consumption rate of first resource type is 49.0.

The average for consumption rate of second resource type is 15.16. All tasks in this project have a standard deviation for consumption rate of second resource type at 10.19. The minimum for consumption rate of second resource type is 1.44. A further investigation of data leads to the fact that the maximum for consumption rate of second resource type is 58.67.

Analysis shows that the average for consumption rate of third resource type is 16.07. The standard deviation for consumption rate of third resource type is 9.75. A further investigation of data leads to the fact that the minimum for consumption rate of third resource type is 1.33. Analysis shows that the maximum for consumption rate of third resource type is 40.0.

The average for consumption rate of fourth resource type is 16.97. All tasks in this project have a standard deviation for consumption rate of fourth resource type at 13.82. All tasks in this project have a minimum for consumption rate of fourth resource type at 0.89. The maximum for consumption rate of fourth resource type is 66.44.

The uncertainty analysis for third project is provided in Table A.12. Analysis shows that the minimum duration mean is 1.67. All tasks in this project have a maximum duration mean at 9.33. A further investigation of data leads to the fact that the minimum standard deviation of duration is 0.58. Analysis shows that the maximum standard deviation of duration is 5.2. A further investigation of data leads to the fact that the minimum scale parameter is 1.9. Analysis shows that the maximum scale parameter is 9.73. The minimum shape parameter is 0.96. All tasks in this project have a maximum shape parameter at 20.71.

A further investigation of data leads to the fact that the average for consumption rate of first resource type is 17.05. All tasks in this project have a standard deviation for consumption rate of first resource type at 10.22. A further investigation of data leads to the fact that the minimum for consumption rate of first resource type is 1.22. A further investigation of data leads to the fact that the maximum for consumption rate of first resource type is 45.33.

The average for consumption rate of second resource type is 17.8. Analysis shows that the standard deviation for consumption rate of second resource type is 11.25. Analysis shows that the minimum for consumption rate of second resource type is 1.11. A further investigation of data leads to the fact that the maximum for consumption rate of second resource type is 47.22.

Analysis shows that the average for consumption rate of third resource type is 17.93. All tasks in this project have a standard deviation for consumption rate of third resource type at 11.77. Analysis shows that the minimum for consumption rate of third resource type is 2.0. All tasks in this project have a maximum for consumption rate of third resource type at 49.0.

A further investigation of data leads to the fact that the average for consumption rate of fourth resource type is 16.98. A further investigation of data leads to the fact that the standard deviation for consumption rate of fourth resource type is 10.76. The minimum for consumption rate of fourth resource type is 1.56. All tasks in this project have a maximum for consumption rate of fourth resource type at 58.67.

4.5 Simulation Output

4.5.1 Tasks Schedule

The main output for any resource scheduling technique is tasks schedule which specifies when each task should be started. Different scheduling techniques will lead to different task schedules. However, there should be a lot of similarities because of the project structure which is assumed to be static. This section analyses the proposed resource scheduling solution with two widely used heuristic schedulers. These schedulers are Longer Task First where the task with longer duration is scheduled first; while the second is Shorter Task first, i.e., the task with the shorted duration is executed first.

The reason behind simulating Longer Task First and Shorter Task First schedulers is to provide a stander benchmarking of comparison. Both of these schedulers are very simple and easy to understand their dynamics which helps when perfuming comparison analysis of proposed scheduler performance. In addition, using three project sizes will give us an indication of how the proposed scheduler performance is affected by project complexity and difficulty of scheduling process. Analysis will be performed for each version of each project.

4.5.2Results Based on Project One

Schedule output for first version of first project is provided in Table A.13.

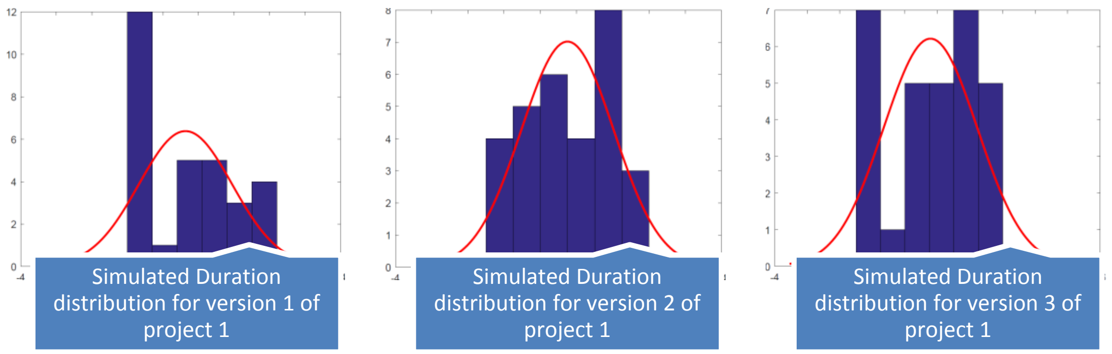


Figure 4.13: Duration distribution of first project.

Table 4.13: Output statistics for version 1 of project 1.

Statistics	Simulated	Longer First	Shorter First

	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
Mean	5.266667	18.06667	23.33333	5.266667	27.13333	32.4	5.266667	27.1	32.36667
Standard Error	0.479303	2.262606	2.119061	0.479303	3.231929	3.031861	0.479303	3.235347	3.035985
Median	5.5	14	22	5.5	27	31	5.5	27	31
Mode	3	4	36	3	8	26	3	8	26
Standard Deviation	2.625253	12.3928	11.60658	2.625253	17.702	16.60619	2.625253	17.72073	16.62877
Sample Variance	6.891954	153.5816	134.7126	6.891954	313.3609	275.7655	6.891954	314.0241	276.5161
Kurtosis	-1.42402	-1.1612	-1.30161	-1.42402	-1.08668	-1.12673	-1.42402	-1.09297	-1.13466
Skewness	0.141995	0.321204	0.134457	0.141995	0.192131	0.127273	0.141995	0.195719	0.130333
Range	8	41	39	8	59	55	8	59	55
Minimum	2	0	4	2	0	6	2	0	6
Maximum	10	41	43	10	59	61	10	59	61
Sum	158	542	700	158	814	972	158	813	971
Count	30	30	30	30	30	30	30	30	30

The average for the difference between proposed and longer first schedulers is 12.33. The standard deviation for this difference is 7.23. A further investigation of data leads to the fact that the minimum for the same difference is 0.0 and the maximum for the same difference is 23.0.

Similarly, Analysis shows that the average for the difference between proposed and shorter first schedulers is 12.3. A further investigation of data leads to the fact that the standard deviation for this difference is 7.21. All tasks in this version of this project have a minimum for the same difference at 0.0. All tasks in this version of this project have a maximum for the same difference at 23.0.

Schedule output for second version of first project is provided in Table A.14.

Table 4.14: Output statistics for version 2 of project 1.

Statistics	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
Mean	5.5	19	24.5	5.5	22.16667	27.66667	5.5	19.93333	25.43333
Standard Error	0.466708	2.317847	2.28224	0.466708	2.352345	2.31702	0.466708	2.395846	2.36482
Median	5.5	17	23	5.5	21	26	5.5	19	23.5
Mode	8	17	17	8	11	26	8	0	18
Standard Deviation	2.556263	12.69537	12.50034	2.556263	12.88432	12.69084	2.556263	13.12259	12.95265
Sample Variance	6.534483	161.1724	156.2586	6.534483	166.0057	161.0575	6.534483	172.2023	167.7713
Kurtosis	-1.07139	-0.7618	-0.66687	-1.07139	-0.75979	-0.66734	-1.07139	-0.75635	-0.63356
Skewness	0.046448	0.331297	0.389605	0.046448	0.333375	0.381754	0.046448	0.311276	0.373109
Range	9	43	46	9	45	47	9	45	48
Minimum	1	0	5	1	2	7	1	0	5

Maximum	10	43	51	10	47	54	10	45	53
Sum	165	570	735	165	665	830	165	598	763
Count	30	30	30	30	30	30	30	30	30

All tasks in this version of this project have an average for the difference between proposed and longer first schedulers at 3.17. All tasks in this version of this project have a standard deviation for this difference at 0.69. Analysis shows that the minimum for the same difference is 2.0. The maximum for the same difference is 4.0.

Similarly, all tasks in this version of this project have an average for the difference between proposed and shorter first schedulers at 0.93. Analysis shows that the standard deviation for this difference is 0.81. All tasks in this version of this project have a minimum for the same difference at 0.0. Analysis shows that the maximum for the same difference is 2.0.

Schedule output for third version of first project is provided in Table A.15.

Table 4.15: Output statistics for version 3 of project 1.

Statistics	Simulated			Longer First			Shorter First		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
Mean	5.5	5.566667	23.13333	28.7	5.566667	27.16667	32.73333	5.566667	24.8
Standard Error	0.466708	0.526719	2.472252	2.519419	0.526719	2.539836	2.592621	0.526719	2.539979
Median	5.5	6	23.5	29.5	6	28	33	6	25
Mode	8	6	37	21	6	15	18	6	38
Standard Deviation	2.556263	2.884959	13.54108	13.79943	2.884959	13.91125	14.20037	2.884959	13.91204
Sample Variance	6.534483	8.322989	183.3609	190.4241	8.322989	193.523	201.6506	8.322989	193.5448
Kurtosis	-1.07139	-1.13062	-0.94622	-0.84378	-1.13062	-0.87598	-0.82522	-1.13062	-0.65917
Skewness	0.046448	-0.13353	-0.09596	-0.04969	-0.13353	-0.04799	0.038744	-0.13353	-0.10675
Range	9	9	47	52	9	50	52	9	50
Minimum	1	1	0	2	1	3	7	1	0
Maximum	10	10	47	54	10	53	59	10	50
Sum	165	167	694	861	167	815	982	167	744
Count	30	30	30	30	30	30	30	30	30

The average for the difference between proposed and longer first schedulers is 4.03. Analysis shows that the standard deviation for this difference is 1.47. A further investigation of data leads to the fact that the minimum for the same difference is 2.0. All tasks in this version of this project have a maximum for the same difference at 6.0.

Similarly, Analysis shows that the average for the difference between proposed and shorter first schedulers is 1.67. The standard deviation for this difference is 1.56. A further investigation of data leads to the fact that the minimum for the same difference is 0.0. Analysis shows that the maximum for the same difference is 4.0.

4.5.3Results Based on Project Two

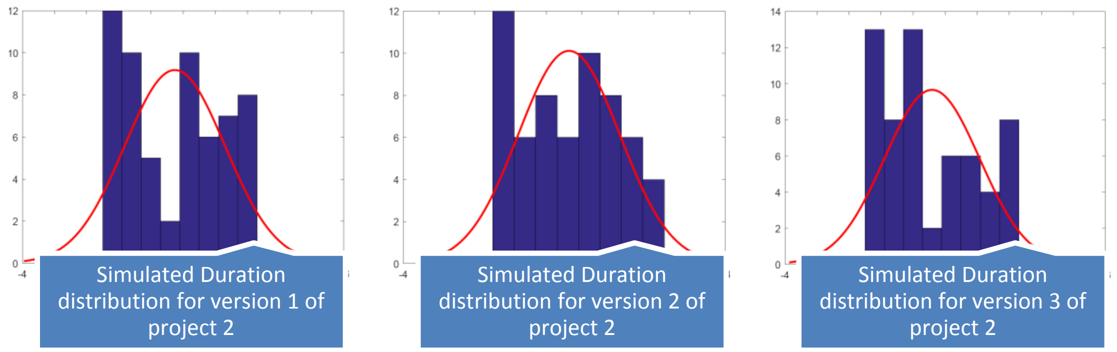


Figure 4.14: Duration distribution of second project.

Schedule output for first version of second project is provided in Table A.16.

Table 4.16: Output statistics for version 1 of project 2.

Statistics	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
Mean	5.483333	29.01667	34.5	5.483333	32.73333	38.21667	5.483333	29.46667	34.95
Standard Error	0.40372	2.467814	2.479008	0.40372	3.067035	3.082674	0.40372	2.524621	2.532771
Median	6	26	32	6	26.5	34	6	26	31
Mode	3	16	19	3	16	19	3	16	19
Standard Deviation	3.1272	19.11561	19.20231	3.1272	23.75715	23.87829	3.1272	19.55563	19.61876

Sample Variance	9.779379	365.4065	368.7288	9.779379	564.4023	570.1726	9.779379	382.4226	384.8958
Kurtosis	-1.41702	-0.10256	-0.21788	-1.41702	-0.34099	-0.45571	-1.41702	-0.19501	-0.29603
Skewness	0.049077	0.576518	0.530507	0.049077	0.677782	0.629087	0.049077	0.562602	0.514756
Range	9	80	82	9	93	95	9	81	83
Minimum	1	0	1	1	0	1	1	0	1
Maximum	10	80	83	10	93	96	10	81	84
Sum	329	1741	2070	329	1964	2293	329	1768	2097
Count	60	60	60	60	60	60	60	60	60

Analysis shows that the average for the difference between proposed and longer first schedulers is 5.18. Analysis shows that the standard deviation for this difference is 5.26. A further investigation of data leads to the fact that the minimum for the same difference is 0.0. The maximum for the same difference is 13.0.

Similarly, Analysis shows that the average for the difference between proposed and shorter first schedulers is 1.52. A further investigation of data leads to the fact that the standard deviation for this difference is 2.6. All tasks in this version of this project have a minimum for the same difference at 0.0. The maximum for the same difference is 14.0.

Schedule output for second version of second project is provided in Table A.17.

Table 4.17: Output statistics for version 2 of project 2.

Statistics	Simulated			Longer First			Shorter First		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
Mean	5.266667	26.43333	31.7	5.266667	29.4	34.66667	5.266667	27.41667	32.68333
Standard Error	0.36659	2.369339	2.376117	0.36659	2.382635	2.389202	0.36659	2.387147	2.396147
Median	5	26	29.5	5	29.5	33	5	27.5	30
Mode	8	8	29	8	11	10	8	30	19
Standard Deviation	2.839591	18.35282	18.40532	2.839591	18.45581	18.50668	2.839591	18.49076	18.56048
Sample Variance	8.063277	336.826	338.7559	8.063277	340.6169	342.4972	8.063277	341.9082	344.4912
Kurtosis	-1.15992	-0.65493	-0.61258	-1.15992	-0.63568	-0.6111	-1.15992	-0.66311	-0.62564
Skewness	0.050752	0.412364	0.323139	0.050752	0.417202	0.334733	0.050752	0.409379	0.328118
Range	9	71	74	9	73	74	9	71	72
Minimum	1	0	2	1	2	6	1	0	4
Maximum	10	71	76	10	75	80	10	71	76
Sum	316	1586	1902	316	1764	2080	316	1645	1961
Count	60	60	60	60	60	60	60	60	60

Analysis shows that the average for the difference between proposed and longer first schedulers is 2.97. The standard deviation for this difference is 0.8. The minimum for the same difference is 2.0. A further investigation of data leads to the fact that the maximum for the same difference is 4.0.

Similarly, the average for the difference between proposed and shorter first schedulers is 0.98. A further investigation of data leads to the fact that the standard deviation for this difference is 0.87. The minimum for the same difference is 0.0. All tasks in this version of this project have a maximum for the same difference at 2.0.

Schedule output for third version of second project is provided in Table A.18.

Table 4.18: Output statistics for version 3 of project 2.

Statistics	**Simulated**			**Longer First**			**Shorter First**		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
Mean	5.183333	23.96667	29.15	5.183333	28.01667	33.2	5.183333	26.3	31.48333
Standard Error	0.383628	2.048834	2.130996	0.383628	2.098088	2.175726	0.383628	2.061799	2.140739
Median	4	23.5	27.5	4	28.5	31.5	4	25.5	30.5
Mode	4	29	11	4	32	42	4	2	18
Standard Deviation	2.97157	15.8702	16.50662	2.97157	16.25172	16.8531	2.97157	15.97063	16.58209
Sample Variance	8.830226	251.8633	272.4686	8.830226	264.1184	284.0271	8.830226	255.061	274.9658
Kurtosis	-1.25215	-0.85506	-0.63671	-1.25215	-0.81976	-0.58337	-1.25215	-0.93151	-0.74057
Skewness	0.388958	0.228671	0.347865	0.388958	0.283306	0.432965	0.388958	0.20117	0.319542
Range	9	62	69	9	65	70	9	61	68
Minimum	1	0	1	1	3	6	1	2	3
Maximum	10	62	70	10	68	76	10	63	71
Sum	311	1438	1749	311	1681	1992	311	1578	1889
Count	60	60	60	60	60	60	60	60	60

All tasks in this version of this project have an average for the difference between proposed and longer first schedulers at 4.05. Analysis shows that the standard deviation for this difference is 1.45. A further investigation of data leads to the fact that the minimum for the same difference is 2.0. Analysis shows that the maximum for the same difference is 6.0.

Similarly, a further investigation of data leads to the fact that the average for the difference between proposed and shorter first schedulers is 2.33. Analysis shows that the standard deviation for this difference is 1.35. The minimum for the same difference is 0.0. Analysis shows that the maximum for the same difference is 4.0.

4.5.4Results Based on Project Three

Schedule output for first version of third project is provided in Table A.19.

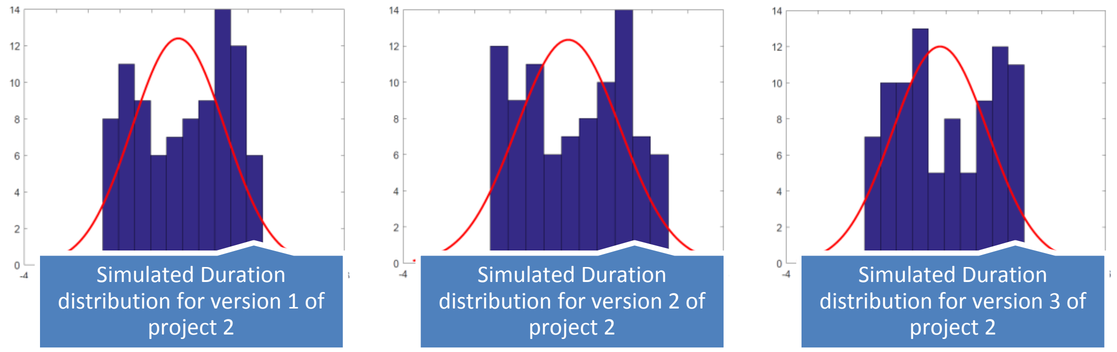


Figure 4.15: Duration distribution of third project.

Table 4.19: Output statistics for version 1 of project 3.

Statistics	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
Mean	5.633333	34.68889	40.32222	5.633333	35.53333	41.16667	5.633333	37.31111	42.94444
Standard Error	0.304953	2.143168	2.127217	0.304953	2.270596	2.232058	0.304953	2.216759	2.196204
Median	6	33.5	40.5	6	34	40	6	37.5	42.5
Mode	8	11	11	8	33	19	8	0	40
Standard Deviation	2.893037	20.33188	20.18055	2.893037	21.54076	21.17517	2.893037	21.03002	20.83502
Sample Variance	8.369663	413.3853	407.2546	8.369663	464.0045	448.3876	8.369663	442.2617	434.098
Kurtosis	-1.34093	-0.86688	-0.88571	-1.34093	-0.80292	-0.80306	-1.34093	-0.76283	-0.74805
Skewness	-0.16571	0.096232	0.056374	-0.16571	0.212813	0.177708	-0.16571	0.028491	0.012665
Range	9	77	78	9	85	86	9	85	86
Minimum	1	0	1	1	0	1	1	0	1
Maximum	10	77	79	10	85	87	10	85	87

Sum	507	3122	3629	507	3198	3705	507	3358	3865
Count	90	90	90	90	90	90	90	90	90

Analysis shows that the average for the difference between proposed and longer first schedulers is 3.8. All tasks in this version of this project have a standard deviation for this difference at 5.21. Analysis shows that the minimum for the same difference is 0.0. The maximum for the same difference is 39.0.

Similarly, all tasks in this version of this project have an average for the difference between proposed and shorter first schedulers at 5.47. Analysis shows that the standard deviation for this difference is 5.73. The minimum for the same difference is 0.0. A further investigation of data leads to the fact that the maximum for the same difference is 26.0.

Table 4.20: Output statistics for version 2 of project 3.

Statistics	**Simulated**			**Longer First**			**Shorter First**		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
Mean	5.277778	30.48889	35.76667	5.277778	33.38889	38.66667	5.277778	31.43333	36.71111
Standard Error	0.306414	1.92456	1.954351	0.306414	1.93494	1.963334	0.306414	1.9285	1.958001
Median	5.5	27.5	32	5.5	30.5	34.5	5.5	28	32.5
Mode	8	15	23	8	18	25	8	15	25
Standard Deviation	2.906899	18.25798	18.5406	2.906899	18.35645	18.62582	2.906899	18.29536	18.57523
Sample Variance	8.450062	333.3538	343.7539	8.450062	336.9594	346.9213	8.450062	334.7202	345.0392
Kurtosis	-1.32696	0.021016	0.011022	-1.32696	0.121827	0.093027	-1.32696	-0.00718	-0.0158
Skewness	-0.03258	0.686314	0.769013	-0.03258	0.706187	0.788836	-0.03258	0.664816	0.752421
Range	9	80	83	9	82	85	9	81	82
Minimum	1	0	2	1	2	4	1	0	4
Maximum	10	80	85	10	84	89	10	81	86
Sum	475	2744	3219	475	3005	3480	475	2829	3304
Count	90	90	90	90	90	90	90	90	90

All tasks in this version of this project have an average for the difference between proposed and longer first schedulers at 2.9. A further investigation of data leads to the fact that the standard deviation for this difference is 0.8. Analysis shows that the minimum for the same difference is 2.0. The maximum for the same difference is 4.0.

Similarly, a further investigation of data leads to the fact that the average for the difference between proposed and shorter first schedulers is 0.94. The standard deviation for this difference is 0.79. All tasks in this version of this project have a minimum for the same difference at 0.0. A further investigation of data leads to the fact that the maximum for the same difference is 2.0.

Table 4.21: Output statistics for version 3 of project 3.

Statistics	Simulated			Longer First			Shorter First		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
Mean	5.633333	40.42222	46.05556	5.633333	44.44444	50.07778	5.633333	42.76667	48.4
Standard Error	0.315021	2.139981	2.122783	0.315021	2.141972	2.126301	0.315021	2.156837	2.144284
Median	5.5	42.5	47	5.5	47	52	5.5	44	49.5
Mode	4	40	40	4	55	46	4	56	43
Standard Deviation	2.988555	20.30165	20.13849	2.988555	20.32053	20.17186	2.988555	20.46156	20.34246
Sample Variance	8.931461	412.1568	405.5587	8.931461	412.9238	406.904	8.931461	418.6753	413.8157
Kurtosis	-1.38855	-0.38204	-0.38792	-1.38855	-0.36158	-0.36615	-1.38855	-0.40038	-0.41001
Skewness	0.038672	-0.05107	-0.0789	0.038672	-0.05951	-0.08902	0.038672	-0.08217	-0.10322
Range	9	89	91	9	91	93	9	91	91
Minimum	1	0	2	1	3	5	1	0	4
Maximum	10	89	93	10	94	98	10	91	95
Sum	507	3638	4145	507	4000	4507	507	3849	4356
Count	90	90	90	90	90	90	90	90	90

Analysis shows that the average for the difference between proposed and longer first schedulers is 4.02. All tasks in this version of this project have a standard deviation for this difference at 1.41. Analysis shows that the minimum for the same difference is 2.0. All tasks in this version of this project have a maximum for the same difference at 6.0.

Similarly, Analysis shows that the average for the difference between proposed and shorter first schedulers is 2.34. The standard deviation for this difference is 1.39. A further investigation of data leads to the fact that the minimum for the same difference is 0.0. A further investigation of data leads to the fact that the maximum for the same difference is 4.0.

4.5.5Resources Availability

An important aspect of any resource scheduling technique is resources availability during project lifespan. A good scheduler will utilize resources as much as possible to increase project efficiency and reduce project time. Therefore, comparing resource scheduling techniques by tracking resource availability over time can provide great insights. At the same time, it shows another way to differentiate among schedulers when they have similar performance in term of project time.

Figures presented in this sections shows the evolution of resources availability as time progress. Here, the x-axis represents time; while y-axis represents accumulated availability of resources. There are four colours which are corresponding to four resources types. As each resource is being consumed by any task in the project, its availability will be reduced. This reduction manifests in figure by reducing the amount of colour corresponding to the resource under usage.

First project resources based on short time scenario :

Resource availability for first project with regard to proposed scheduler is provided in figure 4.15.

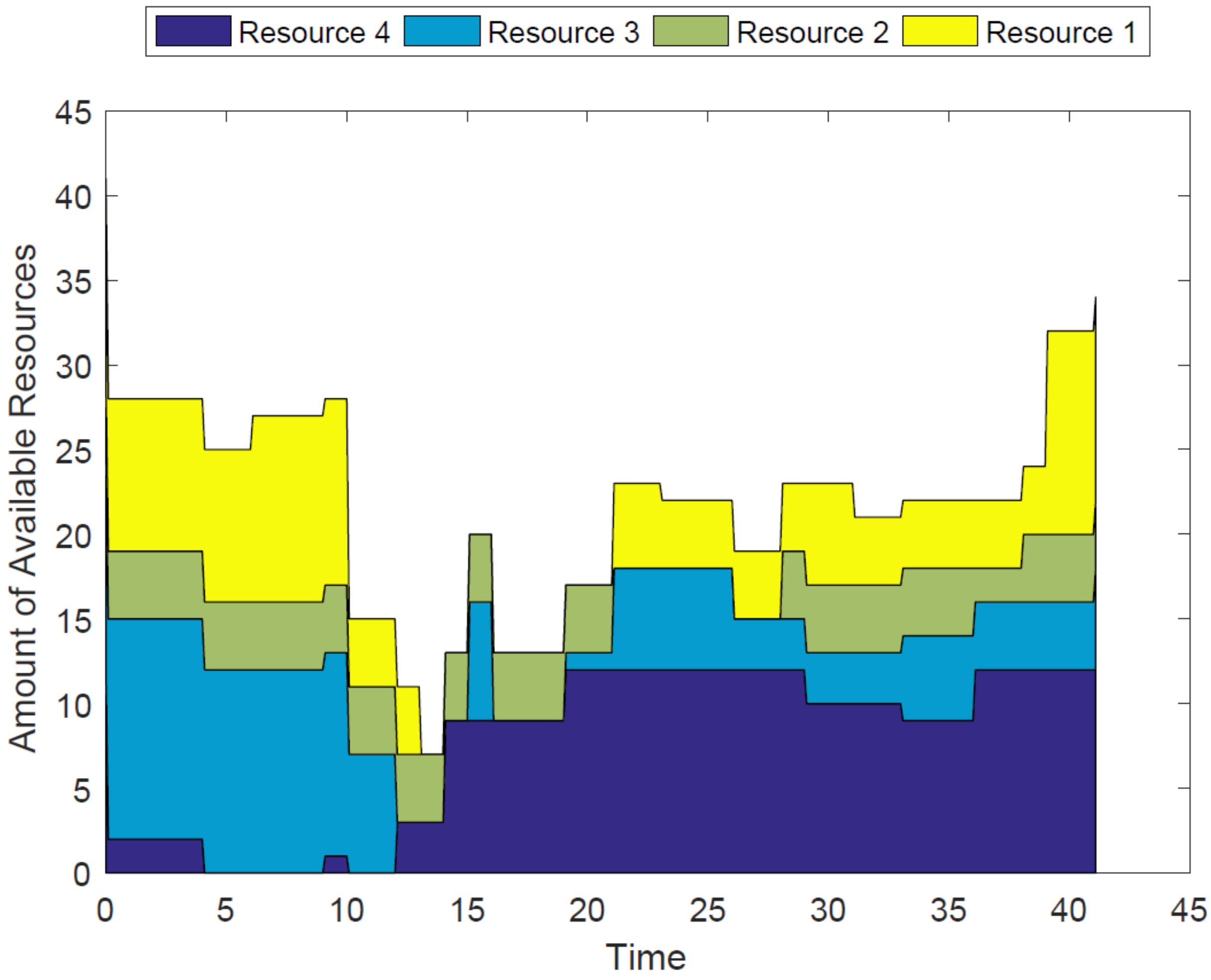


Figure 4.16: Resource availability for first project with regard to proposed scheduler.

First resource type has maximum availability for 2 time units from 19 to 21 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, first resource type has minimum availability for 2 time units from 4 to 6 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Similarly, second resource type has maximum availability for 4 time units from 0 to 4 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 33.0 % of project lifespan. On the other hand, second resource type has minimum availability for 1 time units from 12 to 13 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Third resource type has maximum availability for 4 time units from 0 to 4 at 4 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 33.0 % of project lifespan. On the other hand, third resource type has minimum availability for 2 time units from 21 to 23 at 0 resource units. During this interval, project was consuming 4 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 2 time units from 39 to 41 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 1 time units from 13 to 14 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to fourth resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

First project resources based on long time scenario :

Resource availability for first project with regard to longer first scheduler is provided in figure 4.16.

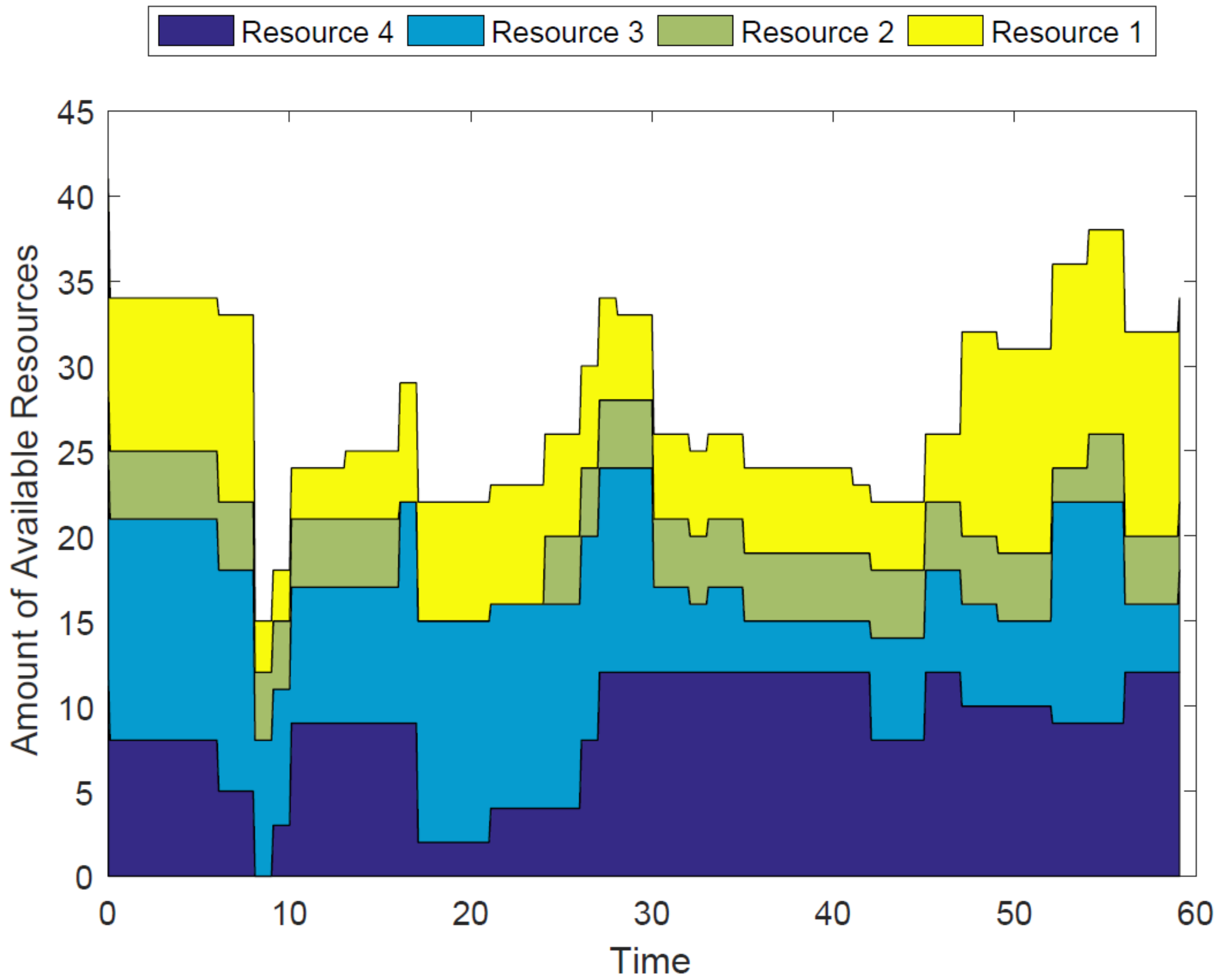


Figure 4.17: Resource availability for first project with regard to longer first scheduler.

First resource type has maximum availability for 1 time units from 27 to 28 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, first resource type has minimum availability for 1 time units from 8 to 9 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Similarly, second resource type has maximum availability for 6 time units from 0 to 6 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 50.0 % of project lifespan. On the other hand, second resource type has minimum availability for 6 time units from 35 to 41 at 3 resource units. During this interval, project was consuming 10 resource units. Its utilization level with respect to second resource type was 77.0 %. This level of utilization was achieved during 50.0 % of project lifespan.

Third resource type has maximum availability for 6 time units from 0 to 6 at 4 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 50.0 % of project lifespan. On the other hand, third resource type has minimum availability for 1 time units from 16 to 17 at 0 resource units. During this interval, project was consuming 4 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 2 time units from 47 to 49 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 1 time units from 8 to 9 at 3 resource units. During this interval, project was consuming 9 resource units. Its utilization level with respect to fourth resource type was 75.0 %. This level of utilization was achieved during 8.0 % of project lifespan. Resource availability for first project with regard to shorter first scheduler is provided in figure 4.17.

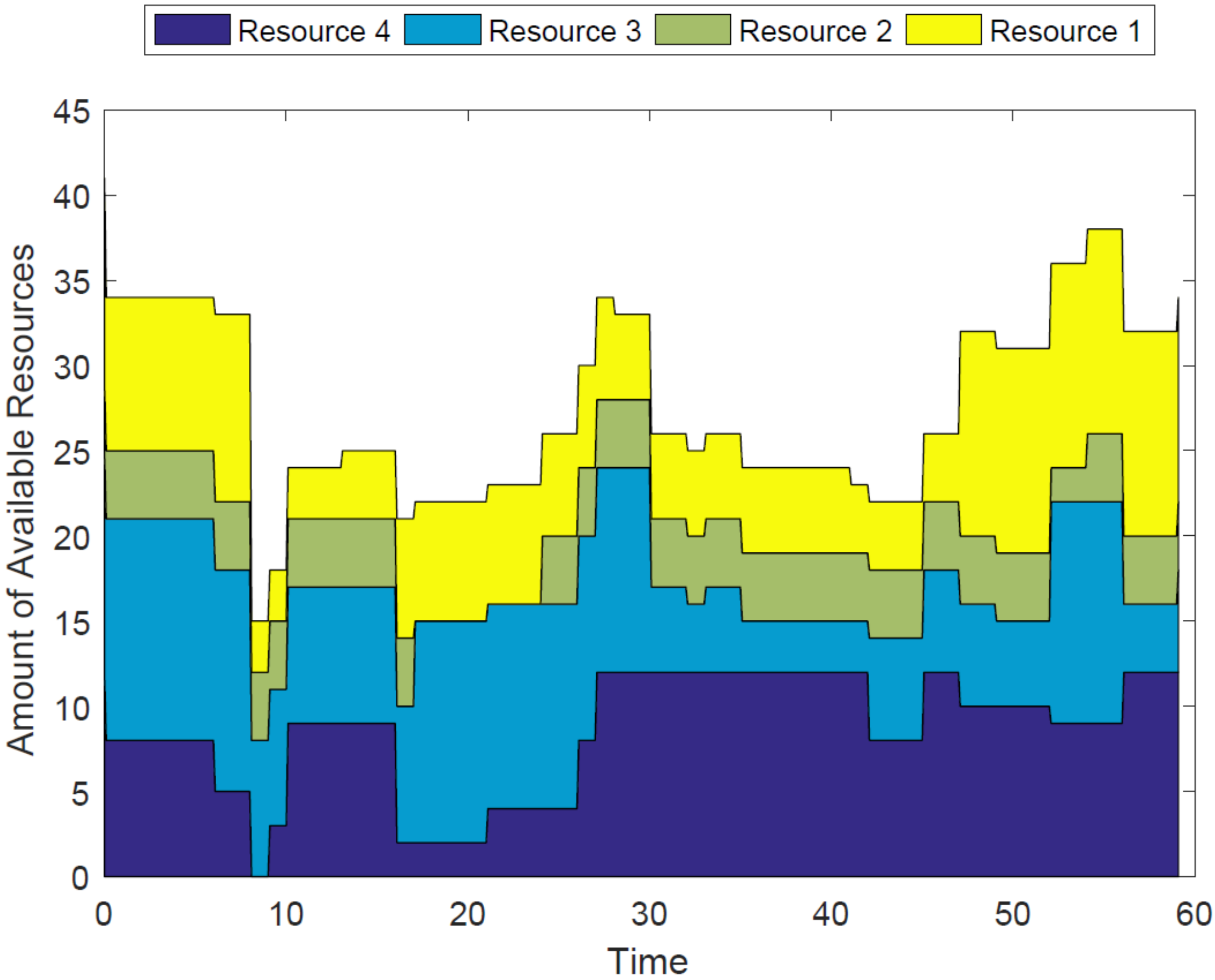


Figure 4.18: Resource availability for first project with regard to shorter first scheduler.

First resource type has maximum availability for 1 time units from 27 to 28 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, first resource type has minimum availability for 1 time units from 8 to 9 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Similarly, second resource type has maximum availability for 6 time units from 0 to 6 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 50.0 % of project lifespan. On the other hand, second resource type has minimum availability for 6 time units from 35 to 41 at 3 resource units. During this interval, project was consuming 10 resource units. Its utilization level with respect to second resource type was 77.0 %. This level of utilization was achieved during 50.0 % of project lifespan.

Third resource type has maximum availability for 6 time units from 0 to 6 at 4 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 50.0 % of project lifespan. On the other hand, third resource type has minimum availability for 4 time units from 17 to 21 at 0 resource units. During this interval, project was consuming 4 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 33.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 2 time units from 47 to 49 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 1 time units from 8 to 9 at 3 resource units. During this interval, project was consuming 9 resource units. Its utilization level with respect to fourth resource type was 75.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Second project resources based on short time scenario :

Resource availability for second project with regard to proposed scheduler is provided in figure 4.18.

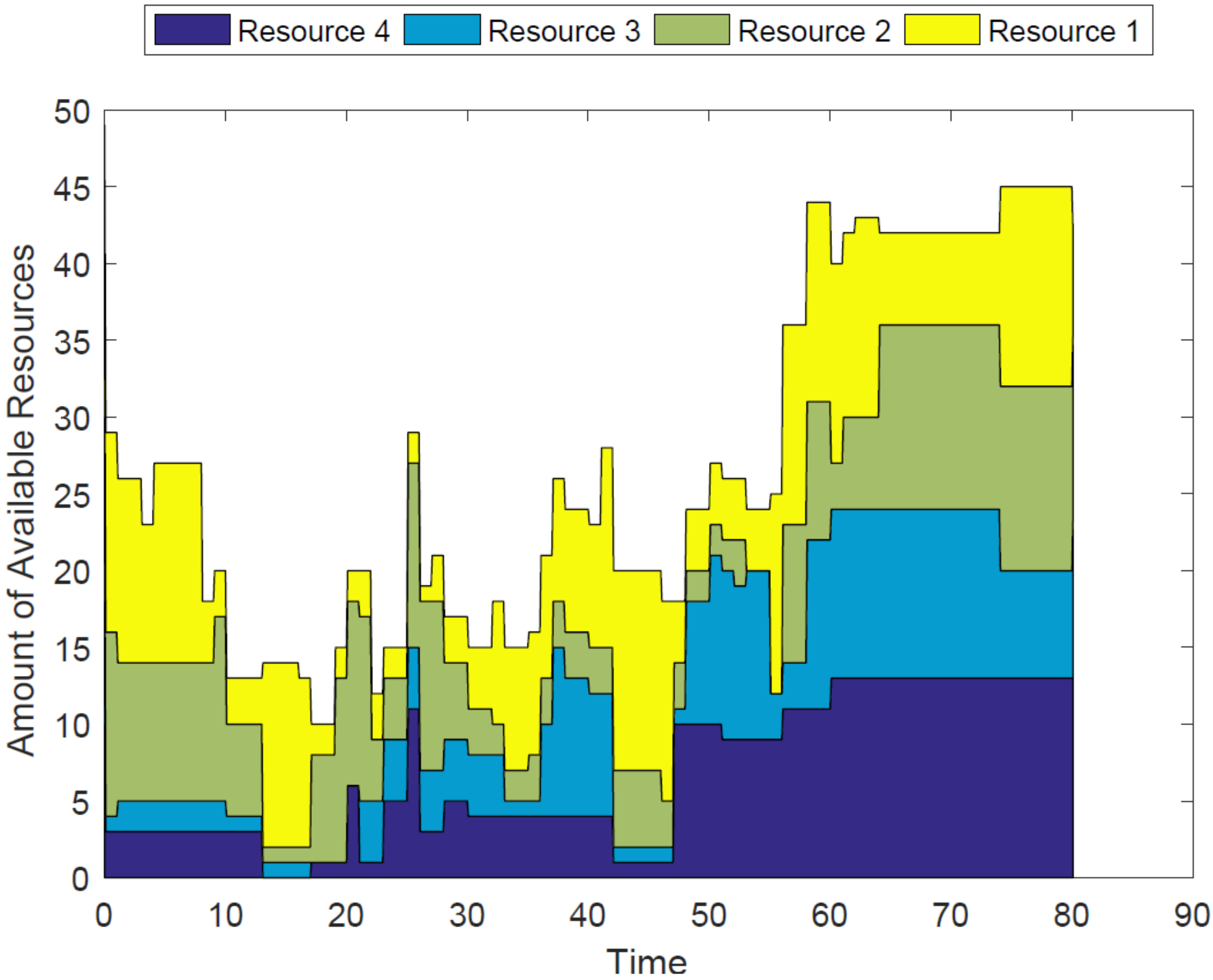


Figure 4.19: Resource availability for second project with regard to proposed scheduler.

First resource type has maximum availability for 1 time units from 60 to 61 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, first resource type has minimum availability for 3 time units from 13 to 16 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Similarly, second resource type has maximum availability for 1 time units from 37 to 38 at 11 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, second resource type has minimum availability for 2 time units from 17 to 19 at 0 resource units. During this interval, project was consuming 11 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 15.0 % of project lifespan.

Third resource type has maximum availability for 1 time units from 0 to 1 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, third resource type has minimum availability for 2 time units from 53 to 55 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 15.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 1 time units from 0 to 1 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 1 time units from 26 to 27 at 1 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to fourth resource type was 92.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Second project resources based on long time scenario :

Resource availability for second project with regard to longer first scheduler is provided in figure 4.19.

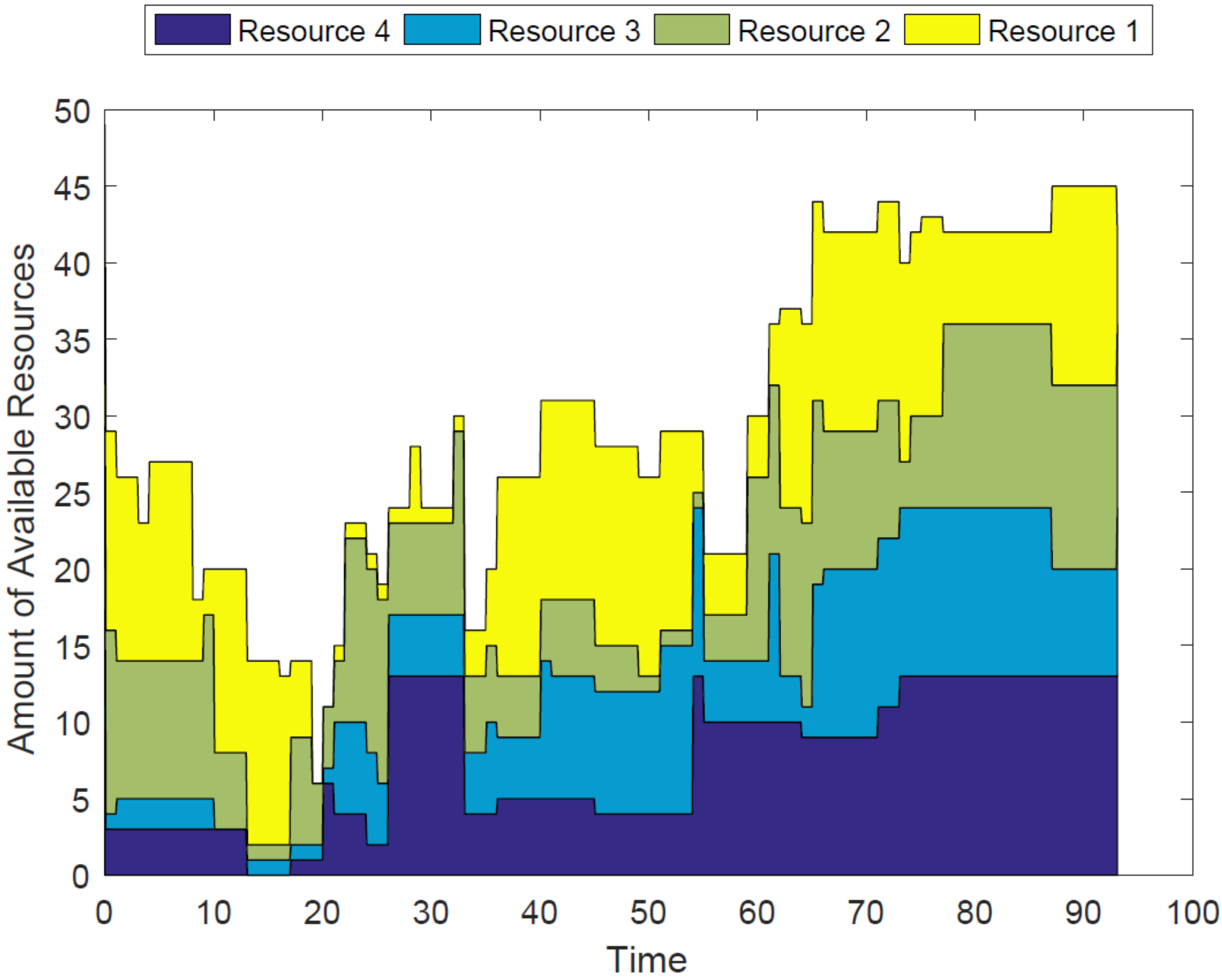


Figure 4.20: Resource availability for second project with regard to longer first scheduler.

First resource type has maximum availability for 2 time units from 26 to 28 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 15.0 % of project lifespan. On the other hand, first resource type has minimum availability for 3 time units from 13 to 16 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Similarly, second resource type has maximum availability for 3 time units from 51 to 54 at 11 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 23.0 % of project lifespan. On the other hand, second resource type has minimum availability for 3 time units from 10 to 13 at 0 resource units. During this interval, project was consuming 11 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Third resource type has maximum availability for 1 time units from 0 to 1 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, third resource type has minimum availability for 3 time units from 13 to 16 at 1 resource units. During this interval, project was consuming 11 resource units. Its utilization level with respect to third resource type was 92.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 1 time units from 0 to 1 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 1 time units from 19 to 20 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to fourth resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Second project resources based on short time scenario :

Resource availability for second project with regard to shorter first scheduler is provided in figure 4.20.

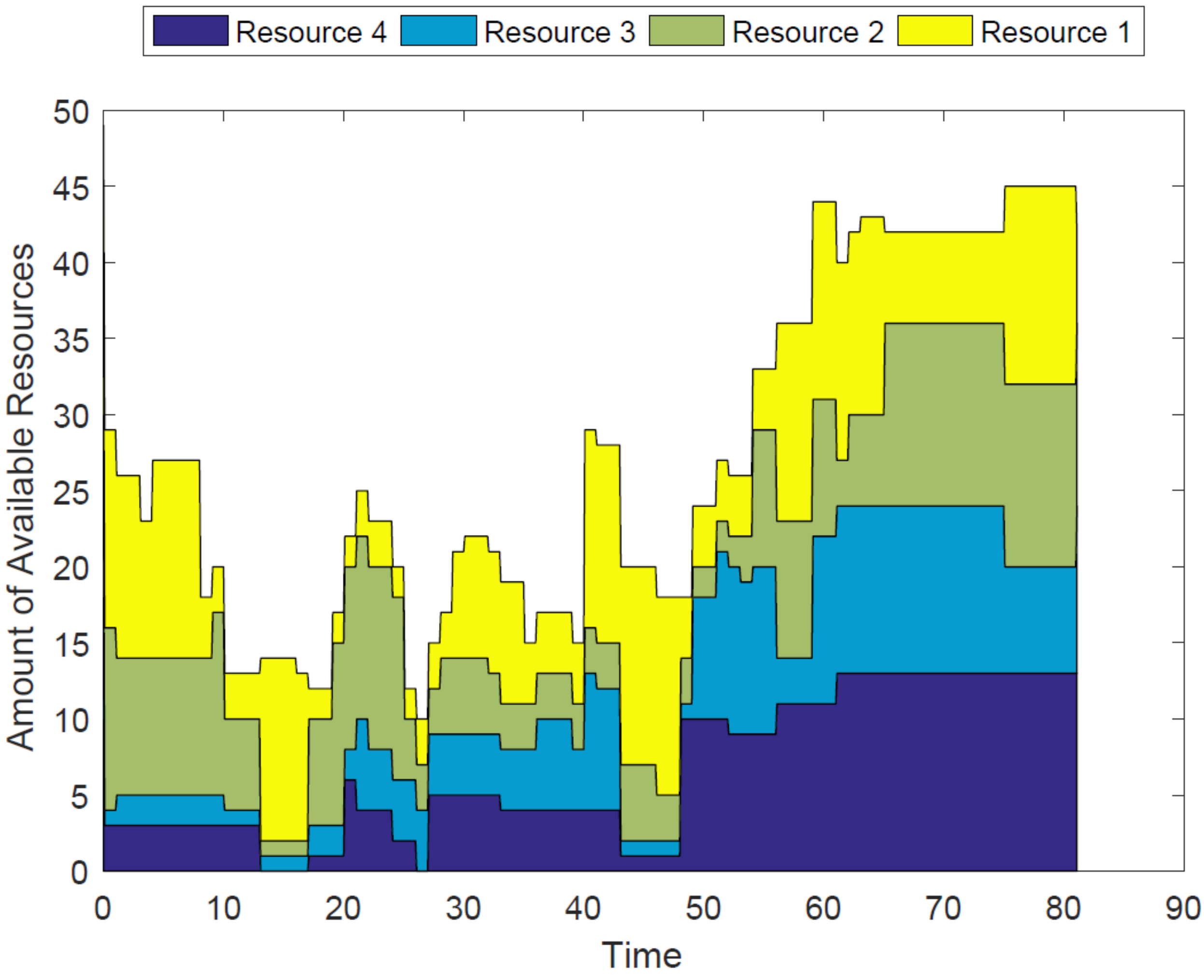


Figure 4.21: Resource availability for second project with regard to shorter first scheduler.

First resource type has maximum availability for 1 time units from 61 to 62 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, first resource type has minimum availability for 3 time units from 13 to 16 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Similarly, second resource type has maximum availability for 1 time units from 51 to 52 at 11 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, second resource type has minimum availability for 1 time units from 0 to 1 at 1 resource units. During this interval, project was consuming 10 resource units. Its utilization level with respect to second resource type was 91.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Third resource type has maximum availability for 1 time units from 0 to 1 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, third resource type has minimum availability for 3 time units from 13 to 16 at 1 resource units. During this interval, project was consuming 11 resource units. Its utilization level with respect to third resource type was 92.0 %. This level of utilization was achieved during 23.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 1 time units from 0 to 1 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 2 time units from 17 to 19 at 2 resource units. During this interval, project was consuming 11 resource units. Its utilization level with respect to fourth resource type was 85.0 %. This level of utilization was achieved during 15.0 % of project lifespan.

Third project resources:

Resource availability for third project with regard to proposed scheduler is provided in figure 4.21.

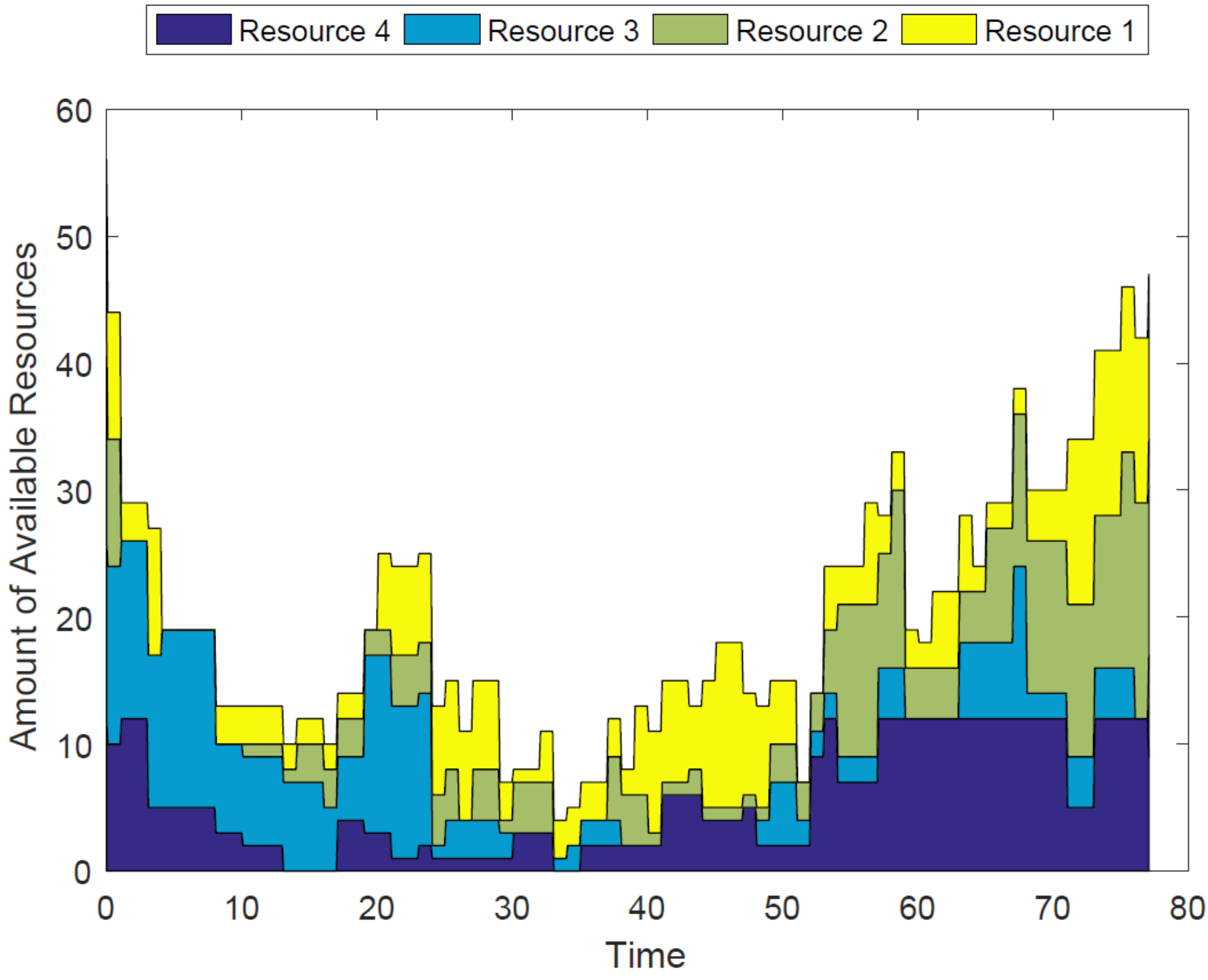


Figure 4.22: Resource availability for third project with regard to proposed scheduler.

First resource type has maximum availability for 2 time units from 1 to 3 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, first resource type has minimum availability for 1 time units from 13 to 14 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Similarly, second resource type has maximum availability for 1 time units from 0 to 1 at 14 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, second resource type has minimum availability for 2 time units from 30 to 32 at 0 resource units. During this interval, project was consuming 14 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Third resource type has maximum availability for 1 time units from 75 to 76 at 17 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, third resource type has minimum availability for 2 time units from 1 to 3 at 0 resource units. During this interval, project was consuming 17 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 2 time units from 45 to 47 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 4 time units from 4 to 8 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to fourth resource type was 100.0 %. This level of utilization was achieved during 33.0 % of project lifespan.

Third project resources based on longer duration scenario :

Resource availability for third project with regard to longer first scheduler is provided in figure 4.22.

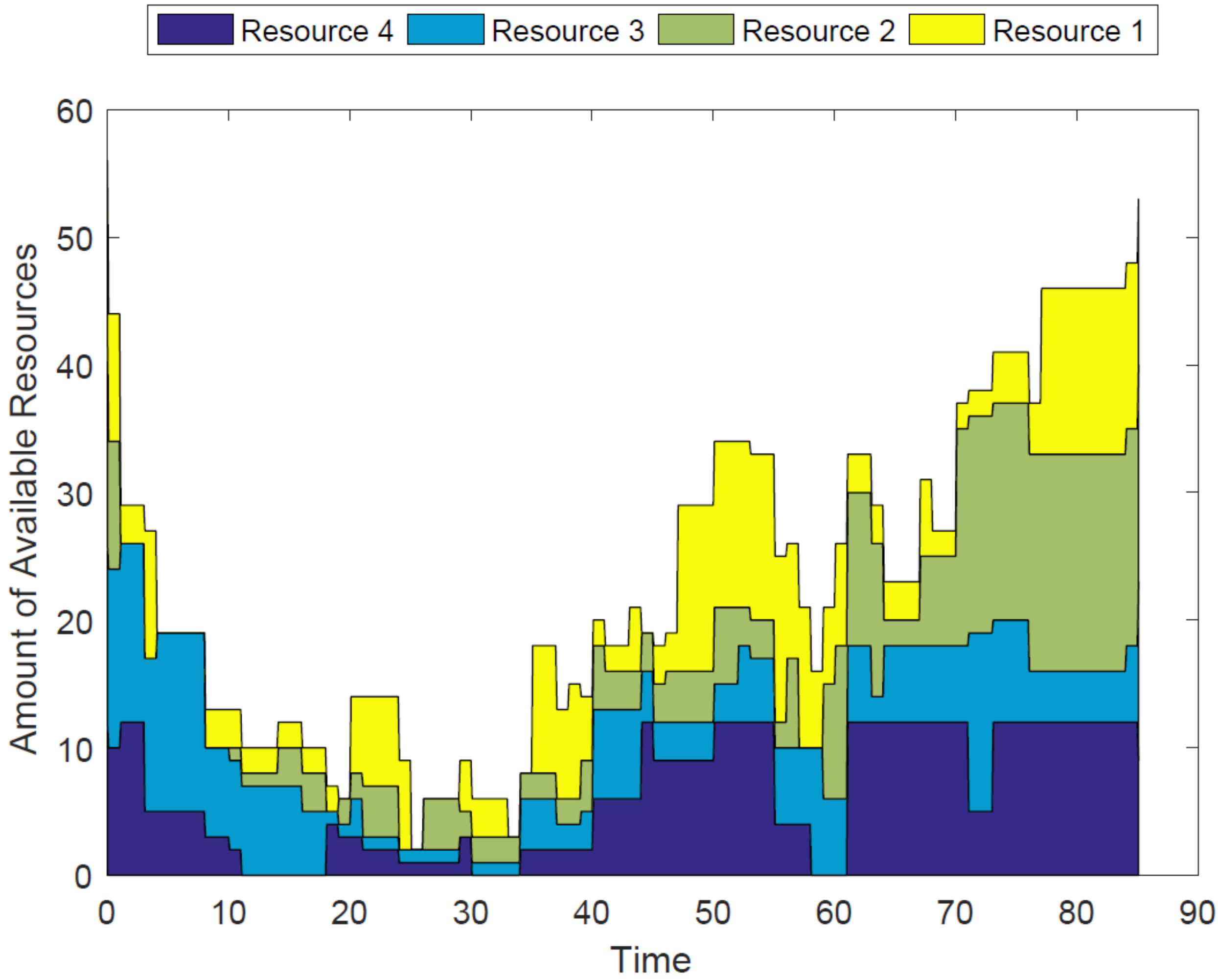


Figure 4.23: Resource availability for third project with regard to longer first scheduler.

First resource type has maximum availability for 2 time units from 1 to 3 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, first resource type has minimum availability for 3 time units from 11 to 14 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 25.0 % of project lifespan.

Similarly, second resource type has maximum availability for 1 time units from 0 to 1 at 14 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, second resource type has minimum availability for 1 time units from 29 to 30 at 0 resource units. During this interval, project was consuming 14 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Third resource type has maximum availability for 1 time units from 70 to 71 at 17 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, third resource type has minimum availability for 2 time units from 1 to 3 at 0 resource units. During this interval, project was consuming 17 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 3 time units from 47 to 50 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 25.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 4 time units from 4 to 8 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to fourth resource type was 100.0 %. This level of utilization was achieved during 33.0 % of project lifespan.

Third project resources based on short time scenario :

Resource availability for third project with regard to shorter first scheduler is provided in figure 4.23.

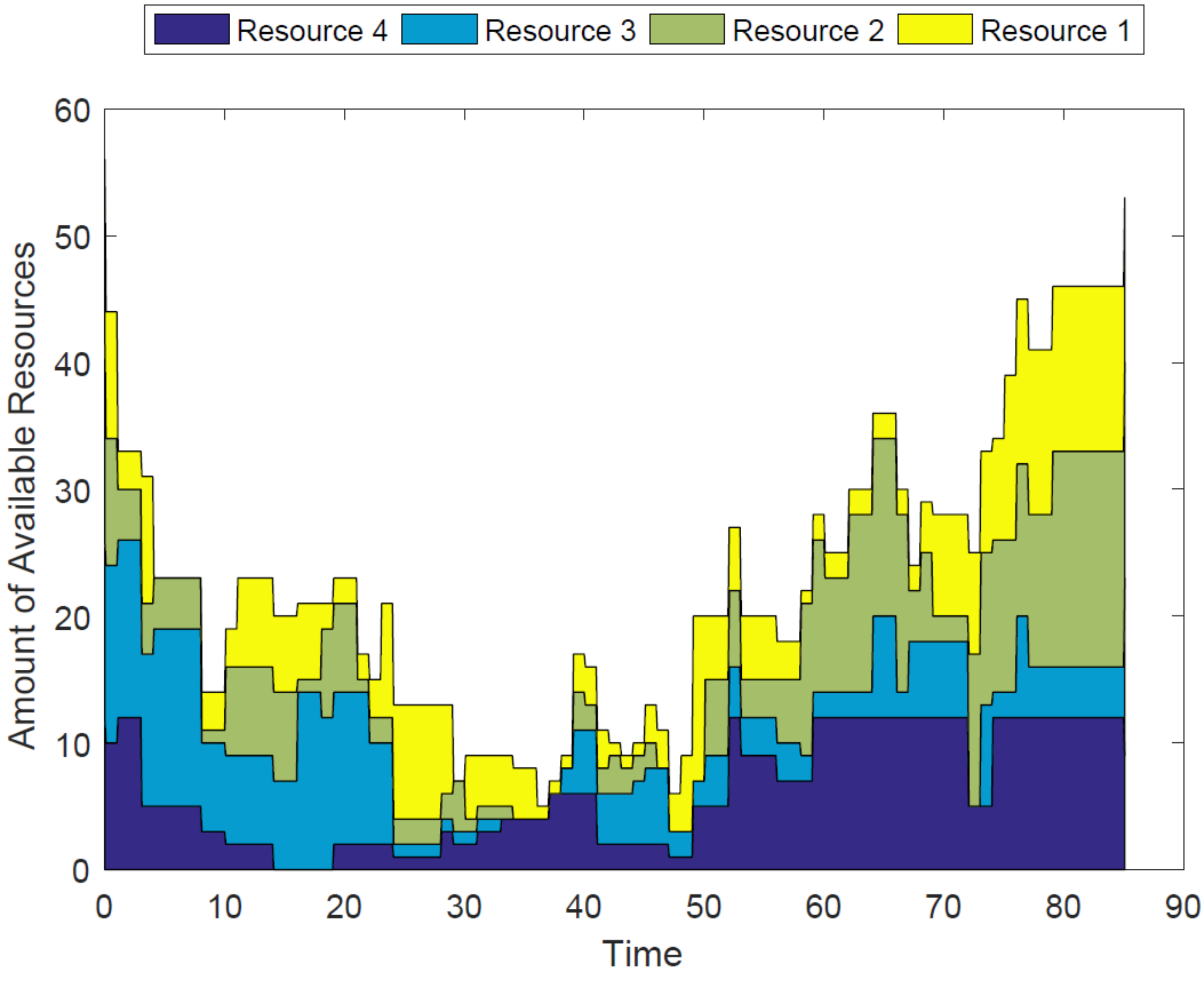


Figure 4.24: Resource availability for third project with regard to shorter first scheduler.

First resource type has maximum availability for 2 time units from 1 to 3 at 12 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to first resource type was 0.0 %. This level of utilization was experienced during 17.0 % of project lifespan. On the other hand, first resource type has minimum availability for 2 time units from 14 to 16 at 0 resource units. During this interval, project was consuming 12 resource units. Its utilization level with respect to first resource type was 100.0 %. This level of utilization was achieved during 17.0 % of project lifespan.

Similarly, second resource type has maximum availability for 1 time units from 0 to 1 at 14 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to second resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, second resource type has minimum availability for 1 time units from 33 to 34 at 0 resource units. During this interval, project was consuming 14 resource units. Its utilization level with respect to second resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Third resource type has maximum availability for 6 time units from 79 to 85 at 17 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to third resource type was 0.0 %. This level of utilization was experienced during 50.0 % of project lifespan. On the other hand, third resource type has minimum availability for 1 time units from 34 to 35 at 0 resource units. During this interval, project was consuming 17 resource units. Its utilization level with respect to third resource type was 100.0 %. This level of utilization was achieved during 8.0 % of project lifespan.

Finally, fourth resource type has maximum availability for 1 time units from 49 to 50 at 13 resource units. During this interval, project was consuming 0 resource units. Its utilization level with respect to fourth resource type was 0.0 %. This level of utilization was experienced during 8.0 % of project lifespan. On the other hand, fourth resource type has minimum availability for 4 time units from 4 to 8 at 0 resource units. During this interval, project was consuming 13 resource units. Its utilization level with respect to fourth resource type was 100.0 %. This level of utilization was achieved during 33.0 % of project lifespan.

4.5.6 Mutation and Crossover Probabilities

The mutation operator is completely random and the resulting output is based upon a number of different choices that are highly random in nature. Ideally the aim of the mutation operator is to create random change that is free from any type of bias. Similar to the mutation function the crossover operator is also completely random and the basis upon which the information is merged or created is completely random. The purpose or aim of the crossover function serves as the main search operator. To illustrate, mutation of 0.1 means that 10 % of the genes in the chromosomes will changed randomly. On the other hand, crossover of 0.7 means that the child chromosome will have 70 % of first parent genes and 30 % of second parent genes.

To begin with the probability for crossover and mutation were set at Pc = 0.5 ~ 0.9 and increased by 0.1, while the probability of the mutation was set at Pm = 1 / N, in this equation the N represents the respective population size of the concerned problem. That’s the reason that this study formulates five distinct alternatives to determine the probability of the input value of mutation and crossover which is then used to analyse and test the algorithm.

4.5.7 Population Size

Setting the population is an important element in the proposed solution as it is utilized in the initial search based upon the initial population identified. It was identified in the preceding chapters that if there are fewer individuals, the GA mechanism would have far fewer opportunities of instilling a crossover resulting in the exploration of only a limited space. In other scenario, if there are too many individuals, the GA would be significantly slowed down. It was presented that an increase or multiplication in the population can be detriment to the algorithm as it increases the length and complexity of the problem delaying the solving process.

Thirty is an acceptable number for the initial population size but in previous studies it was suggested that the lowest population size should be used. But there have been numerous studies that have utilized a higher population size of about 50 and/or 100. The population size should therefore be chose with careful consideration to ensure that it has ample diversity while not increasing the complexity of the problem ensure optimum performance and perfect implementation of the proposed solution.

To determine an appropriate population size for GA, this study elected different project instances with a range of tasks (30, 60, and 120) from the PSPLIB. To run the GA, the size of the population was fixed at 30, 50 and 100 respectively to acquire the initial starting generation that combines to form a solution with a deviation of zero. The probability of crossover was fixed at 0.5 while the mutation was fixed at .03 (1/30), 0.02 (1/50), and 0.01 (1/100) regarding the J30, J60, and J120. Each set or group of instances has 30 distinct problems and 4 resources that are renewable.

It was clear that the different between the fitness values if almost non-existent despite the changes in population sizes from 30 to 100. But it is important to note that the generation averages grow in size due to the increase in tasks or population size. It was evident that due to the increase in task sizes, the generation that converged took long time to obtain. The tests also revealed that the size of population for GA is not dependent on the tasks size. Therefore it can be set at the level of 30 for GA to achieve computational efficiency. It can be also set between the value of 50 or 100 to assist GA in finding the optimum solution space because GA employs the population size for its initial search.

4.5.8 Design of Experiments

The study also evaluated the design through experiments to assess the performance and abilities of the proposed solution. The 3 distinct instances obtained from PSPLIB were employed for the purpose of this experiment. The instances had about 30, 60, and 90 tasks. A full factorial design was employed for the construction of parameters, these parameters outline the different characteristics of available resources and also set the limits for different constraints. Each problem was assigned with 4 distinct renewable resources. The different set of 30 tasks (J30), 90 tasks (J90) and 60 tasks (J60) has 480,600 and 480 different instances respectively. Combined together there were about 1560 different instances. Each set of instances had 4 renewable resources.

The set of standard instances were obtained from PSPLIB by changing three parameters in form of resource strength, network complexity, and resource factor. Network complexity represented the immediate successors for a concerned activity. Consider an example where the resource factor set at zero establishes that the problem in question does not have any resource requirements therefore the resource restrictions are absent. Resource factor represents a situation where each activity requires a single dedicated resource. It also referred to as capacity factor. For example if the capacity factor is set at zero its represents that the resource's capacity is identical to maximum size of resource to ensure feasibility, but in scenarios where the capacity factor has been fixed at 1 it means that there would not be any conflict for the resources. As far as the complexity of any problem is concerned then these three factors have huge significance.

4.5.9Performance Comparison

This section primarily deals with the level of performance achieved by the proposed solution when compared to solutions in literature through the best fitness value identification, which can also be dubbed as the near-optimal result or solution for the scheduling problem. Here, proposed solution is the scheduler developed in Chapter 3. Both of the proposed scheduler and the one from literature (Goncalves, 2008) are utilizing GA. However, Goncalves scheduler has greatly modified GA to a point where its performance is affected. Goncalves needed this modification since modelling project scheduling is not straight forward under GA methodology. The proposed scheduler address this issue by proposing new encoding scheme which based on simulation paradigm as discussed in Chapter 3.

Incorporating uncertainty is one of the main different between these two algorithms. The instances of standard nature gained through the PSPLIB were utilized for this test. The values for optimization were set in the following manner. Mutation probability, initial population size and crossover probability were all set at 0.03, 100 and 0.5 respectively.

Table 4.22: Project life span evaluation for project with 30 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	45.4494	46.1964	50.6299	56.7748	63.2697	69.6781
Literature (Goncalves, 2008)	45.4494	48.4652	56.1401	65.2583	75.0293	92.7374
Heuristic (Longer First and Shorter First)	49.8163	51.4691	63.1433	74.9594	91.9033	115.6752

The results concluded that classical GA proved to be inferior to the proposed scheduler when discovering fitness values. Both algorithms were able to render improved results as compared to PSPLIB besides J30. The proposed solution was able to present an improvement on average of 9.51%, 10.82%, and 12.75% for J30, J60, and J90, respectively compared to GA schedulers in literature. It was also revealed that as the problem grew more complex the ability of the proposed scheduler to represent better results.

Table 4.23: Project life span evaluation for project with 60 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	58.1492	82.5562	89.8604	96.8190	104.1763	113.2705
Literature (ref here)	58.1492	86.4519	99.0742	109.4621	125.3377	140.9795
Heuristic (Longer First and Shorter First)	75.8241	91.8122	109.3486	123.7409	148.6974	173.7134

The proposed was able to represent better life span value in merely around 4600 generation but on the other GA required almost 5700 generations. The proposed scheduler was able to identify improved solutions at a slower pace compared to literature in terms of computational time due to incorporation of uncertainty calculations. However, this incorporation ability represented improved results when the solution was discovered.

Table 4.24: Project life span evaluation for project with 90 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	76.2595	97.9135	107.8800	115.9891	124.1130	133.9418
Literature (ref here)	76.2595	103.3445	116.6282	134.5545	155.4806	165.3613

Heuristic (Longer First and Shorter First)	98.7846	107.7373	128.0817	155.9567	186.2155	213.8874

Further different analyses were also formulated to test the proposed solution behaviour based on the crossover probability as well as the population size. The study revealed that the proposed scheduler continued to present consistent results despite variations in the combinations for RCPSP. But despite this it was presented that as the population size multiplies, the proposed scheduler presents improved solutions.

It was also noted that as the crossover probability was increased the proposed scheduler tendency to follow different patters compared to literature solutions. Therefore it is safe to predict that population size selection plays a more critical role than crossover probability regarding the proposed scheduler.

It was presented that the proposed scheduler continued to require identical computational times in order to discover a fitness value in a RCPSP issues with nominal differences for the CPU times irrespective of the population size. It was also discovered that the increase in crossover probability is most likely to demand lesser computational time to discover a fitness level in a RCPSP scenarios. As a result it can be insinuated that election of higher crossover probability decreases the computational time required for the proposed scheduler.

Table 4.25:Project average start time evaluation for project with 30 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	18.4238	19.0112	19.3215	19.9183	20.3173	20.8352
Literature (ref here)	18.4238	20.0880	21.1923	22.5122	23.6639	25.7809
Heuristic (Longer First and Shorter	20.6758	21.2014	23.1225	26.2290	29.5137	32.6143

First)						

The results depict that the proposed solution continued to present consistent results with nominal differences in the fitness values irrespective of the varying combination used. It was also presented that the increase in population size does not guarantee better results. The proposed scheduler also presented varying results as the probability for crossover increased. Therefore it is insinuated that the selection of population is a very critical factor at J30 as compared to probability of crossover.

Table 4.26: Project average start time evaluation for project with 60 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	28.3475	32.5444	33.2800	34.1379	35.0423	36.0620
Literature (ref here)	28.3475	34.2772	36.3651	39.4255	41.9007	43.6569
Heuristic (Longer First and Shorter First)	33.6478	36.1693	39.7945	45.0578	51.1650	53.4625

The results concluded that the proposed solution demands identical computation time to acquire fitness values in a RCPSP problem despite the increase in population size. They also suggested the different behaviour of the computation runtime based on the varying crossover probability. It was represented that as the probability of the crossover increases, the proposed scheduler would continue to demand identical computational times to acquire a fitness value in a RCPSP scenario.

It was discovered that the proposed solution continues to provide consistent results in a RCPSP scenario with nominal variations among the fitness level averages irrespective of the varying combinations used. But despite that it was also noted that the increase in population size, the proposed scheduler cannot guarantee optimum or improved solution.

Table 4.27: Project average start time evaluation for project with 90 tasks with 95 % confidence.

Schedulers		Uncertainty				
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	36.6738	38.7492	39.8203	40.8043	42.0897	43.1678
Literature (ref here)	36.6738	40.7513	43.8077	46.3097	51.3606	54.0518
Heuristic (Longer First and Shorter First)	40.4684	42.5321	47.9662	53.9626	63.2369	71.3015

It was revealed that the proposed scheduler tended to require a longer computational time to acquire a good fitness value in a RCPSP problem as the population size increased. It was concluded that as the probability for the crossover increased the proposed scheduler requires different computational times to acquire a fitness value in a RCPSP scenario irrespective of the crossover probability.

It is critical to note that only the optimum performing heuristics were taken into consideration. The proposed solution continued to present consistent and accurate results in RCPSP scenarios with nominal differences between the fitness value averages resulting in comparable.

Table 4.28: Project average finish time evaluation for project with 30 tasks with 95 % confidence.

Schedulers	Uncertainty					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	26.2465	27.6494	26.8941	26.1738	25.5168	24.8223
Literature (ref here)	26.2465	29.1999	31.4198	31.2449	31.2215	31.8059
Heuristic (Longer First and Shorter First)	29.7134	30.6192	34.9468	35.1132	36.8439	39.5845

The most important scheduling objective for any practitioner or researcher is the total life span of the project. Usually, minimizing life span is the main concern of project managers. Other concerns may play roles such as resources utilization or total project cost.

Hence, most research efforts on resources scheduling try to address this preferences in their proposed solution. The main evaluation effort of this research is to find out how life span of project is minimized in comparison to other works in literature and widely used heuristic solution. These heuristic solutions are usually utilized by practitioners such as project manager due to their ease of usage.

Table 4.29: Project average finish time evaluation for project with 60 tasks with 95 % confidence.

Schedulers		**Uncertainty**				
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	45.1763	46.8648	45.6364	44.3196	43.3051	42.1575
Literature (ref here)	45.1763	49.8190	52.6628	52.5809	54.5771	55.9552
Heuristic (Longer First and Shorter First)	51.7446	52.0140	56.8070	60.3845	66.1304	68.8235

Nevertheless, other performance metrics can shed a light on important aspect of proposed scheduler performance. For example, both average start time of projects tasks and average finish time of projects tasks highlight how proposed solution handle increased complexity. As level of uncertainty increases, the average start time of project tasks keeps almost constant for the proposed solution; while it keeps increasing for other solutions. On the other hand, the proposed solution was able to utilize the increased scheduling complexity as shown in average finish time evaluation. The keeps reducing average finish time of project tasks as complexity and uncertainty increase. The proposed scheduler was able to find scheduling decisions that push executions of tasks so that they finish as early as possible.

Table 4.30: Project average finish time evaluation for project with 90 tasks with 95 % confidence.

Schedulers	**Uncertainty**					
	0 %	**5 %**	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	52.4325	56.8676	55.6420	54.4416	52.9623	51.5392
Literature (ref here)	52.4325	60.3734	64.6801	66.2134	66.4510	69.3377
Heuristic (Longer First and Shorter First)	59.7231	63.5651	71.9388	73.2647	78.3833	89.1423

4.6 Applicability Evaluation Phase

Any good resources scheduling mechanism has no value if it cannot be used in real project portfolio management environment. Therefore, one of the most important phases in this research is to study how these proposed technically advanced mechanisms can be adopted in real situations. This phase was conducted by using real project from Al-Ain municipality in UAE. This project was chosen from IT department. It was about developing mobile application for human resources management. The development process is consisted of several tasks such as Interface Design, User Experience Evaluation, Database Integration, Back-End Server Configuration and other. For the sake of simplicity will denote these tasks with numbers with worrying about their exact natures since they don't play any role in scheduling process. There is only one renewable resource which is programmers. The project was assigned six programmers in total and they are going to perform all projects tasks. The following diagram describes project structure.

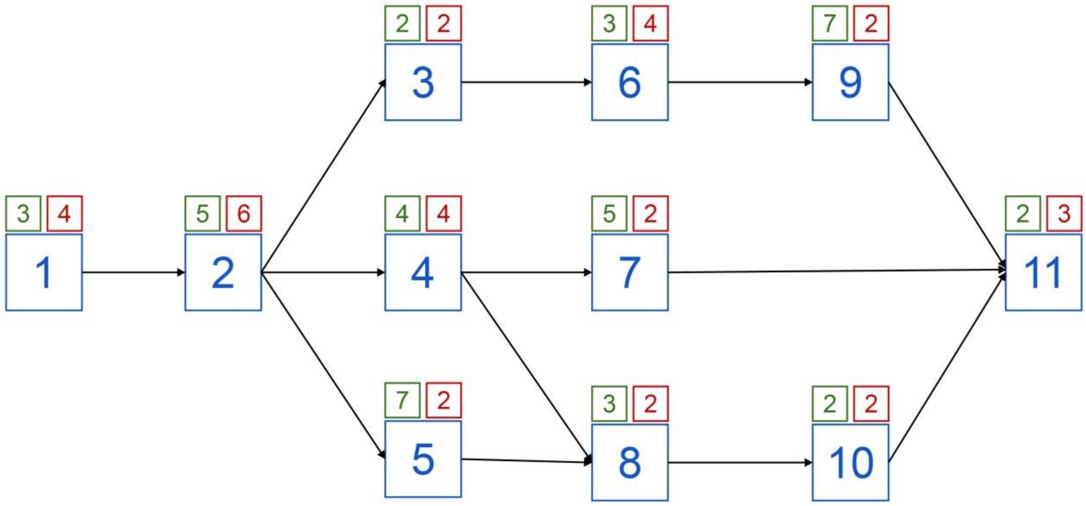


Figure 4.25: Mobile application development project with expected time for each task.

Each task has a number describing its index which is the number in blue in Figure 4.24. Also, there are two numbers describing tasks duration and amount of required resources. The task duration in days is represented by green number; while red number represents amount of required resources which are the number of needed programmers to complete the task. Note that task duration numbers in Figure 4.24 are the estimation at the beginning.

There are 11 tasks in this project which different durations, resources requirement and dependency on other tasks. We need to schedule these tasks in a way that minimize life span of the project. Project manager used Longest Task First rule which led to this scheduling.

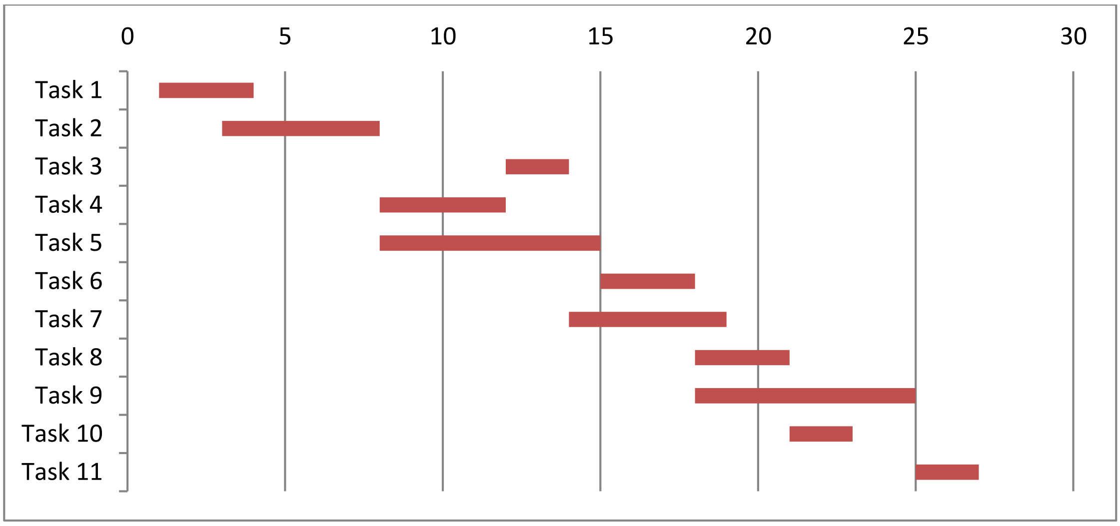


Figure 4.26: Gantt chart of development project scheduling.

Based on this scheduling, the project life span is 27 days. However, it took them 39 days to complete the project. The reason behind discrepancy is the unreliable estimation of tasks durations. Some tasks took longer time than expected; while others finished earlier. Due to the fact that programmers are also working on other projects and cannot start working on any task out of the schedule, this led to wasting a lot of time rescheduling tasks from different project to address wrong estimation of tasks durations.

Table 4.31: Life span of development project when each of the considered schedulers are used.

Schedulers	**Uncertainty**				
	5 %	**10 %**	**15 %**	**20 %**	**25 %**
Simulated	30	31	33	34	36
Heuristic (Longer First and Shorter First	39	39	39	39	39

Considering uncertainty regarding task duration while performing project scheduling can lead to more reliable scheduling decisions. Figure 4.25 compares the life span of project development if each one of considered approaches are used in scheduling process. It is clear from the figure that using the proposed approach will result in shorter life span. Both classical genetic algorithm approach and heuristic approach have the same performance because the project is small and doesn't have a lot of complexity. Figure 4.26 shows resource utilization throughout project time. There has been at least utilization of three resource units. In some periods, all available resource are fully utilized.

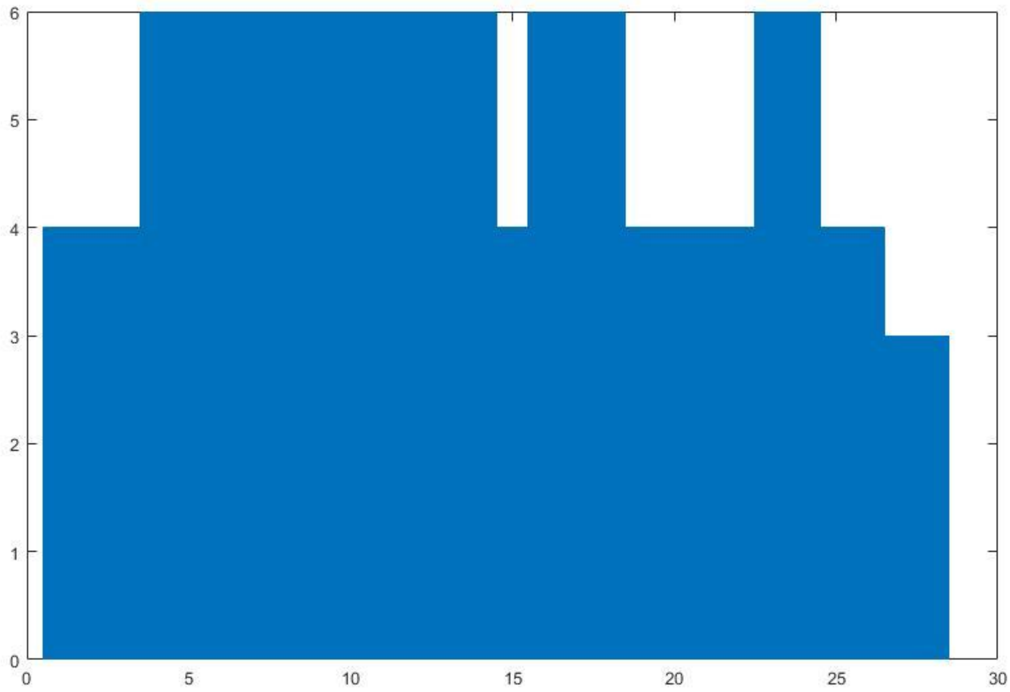


Figure 4.27: Resource utilization.

4.7 Chapter Summary

This chapter showed performed extensive evaluations of the proposed scheduling solution in chapter 3. Two sets of evaluations were conducted. The first set was based on well-known benchmark of scheduling problem which widely used in literature. The second evaluation was based on real scheduling problem for mobile application development project. All evaluations concluded that the proposed approach is superior in term of minimizing project life span compared to other scheduling solutions in literature.

Chapter 5: Discussion and Conclusions

5.1 Introduction

The main purpose of the last chapter of this thesis is to interpret and explain generated results in comparison with the existing studies where validity of adopted approach is examined. The following sections will provide an overall view of research issues and activities in addition to main research findings and conclusions.

5.2 Overview

As mentioned earlier, project management refers to a coordinated planning and control effort through which all the activities of a project are directed or monitored. Project management deals with the allocation of resources, achievement of objectives, and timing of certain actions and finalization processes that would be required to complete a project successfully. Project management provides a direction for the team and is essential for the successful completion of the project. As per the definition of Project Management Institute, project management is recognizing the different requirements of the project, creating clearly identifiable objectives and allocation of resources in a just way that ensures that the maximum interests of all stakeholders are met. A successful project management requires that a project is completed within the estimated time and consumes the resources effectively in accordance with the wishes of the client to achieve different objected that were intended out of the project.

From the perspective of management a project has five unique stages that involve different management actions in order to successfully achieve its objectives. First stage of a project is called project conception, where there is a lack of concrete information and through the feasibility studies and other means the management decides whether the project should be undertaken or not. At the second stage of the project called Project Definition phase the project's goals, organization and objectives are clearly outlined. In addition to the goals this phase also defines the timeline of the project in form of a roadmap. At the next stage called the Project Planning Phase the project is broken down into smaller activities that would be required to complete the project, then for each activity the resources are allocated in form of Time, manpower and material and an estimation of the cost associated with the execution of these activities. Furthermore any dependencies for each activity are defined and a finalized schedule is created for the project. These three initial phases concentrate on the planning and resource allocation for the project after which the project finally enters into its execution phase. The management constantly monitors the project in order to determine whether the project is operating as planned or some variances have risen that require corrective actions. Furthermore the project is also evaluated in term of the quality that helps the management understand the performance level of the project. At the final stage called Termination phase the project is document and finally reaches its conclusion.

This thesis focused mainly on project scheduling which deals with the methods of execution for each activity and the time frame for each activity is defined as well. Projects may be faced with scarcity of resources either in form of time or scarcity of other resources such as manpower and finance. In case of limitation of time the management assumes that the activities cannot be performed at the current financial requirements and the project would need to incur higher costs in order to achieve the desired output. While in the resource constraint the organization does not have the required amount of resources or has a limited amount of resources therefore it is extremely important to take resource constraints into account as well along with the time constraints.

The term “resource” in this thesis refers to the pool of identical amount of resources and resources availability is referred in form of capacity of resource. At the project scheduling stage several different kinds of resources are allocated based on the availability and requirement of the activities. Renewable resources refer to a resource that does not lose its value or ability to add to value to the organization after the initial application, Manpower and machinery are common examples of renewable resources. While non-renewable resources are consumed or lose their value during the application in the project Raw Material and budgets are common example of non-renewable resources. Renewable and non-renewable resources are usually consolidated in a single category called storage resource. The unit of storage provides a more accurate and easily identifiable measure that can be used to determine the cash balance for the project with regards to the distribution and progress of the payment. Resources such as electricity and virtual memory of the computers and other machine systems can be allocated in a continuous stream are called continuous resources. Partially renewable resources refer to the merging of different time intervals in order to estimate the labor requirement for a project.

A common assumption used in the most projects is that the activities cannot be stopped or interrupted during their execution. However, this is not a common situation while managing real projects. It is usual for a manager to halt any task before it is completed to use its resources in another task. For example, an employee may be working on specific task and his manager will asks him to stop what he is doing to perform another task even if the first one is not completed yet. It is clear that no one can anticipate beforehand if any task will be interrupted or not. Therefore, this assumption is necessary so that tackling scheduling problem can be feasible. An important aspect of problem scheduling deals with the different executions options that are available to the management for a certain project. Single-Model problem assumes that the management does not possess the luxury to select the different execution models or an appropriate execution model has been selected that cannot be changed. While in a multi-model scheduling problem the management can consider several different alternatives to complete an activity this model enables the organization to consider different resource and cost based trade-offs that are a common occurrence in the modern economic environment.

With regards to the scheduling objective the management first needs to differentiate between single-criterion and multi-criterions scenarios. In a multi-criterion problem the management is required to use attainment or goal programming models in order to achieve an acceptable solution. The second function can be distributed between regular and non-regular objectives. Regular objective functions are non-decreasing component driven functions carried out at the initial stages of execution phase of activities. In such a situation the activities cannot be started without compromising the time and resource allocation of other activities. That is the sole reason that problems associated with the regular function can be more easily dealt with as compared to the non-regular functions.

5.3 Literature Review

The first phase in this research was conducting an extensive literature review regarding resources scheduling in project management. From the beginning of review process, it was clear that uncertainty issue is very important to have effective resources scheduling. Hence, it was chosen as the main focus of this research. According to performed literature review, uncertainty models are used to convert the inputs to outputs. It is of vital importance to incorporate uncertainty measure into a model that would increase the efficiency of the overall results. There are two broad types of uncertainties called aleatory uncertainty and epistemic uncertainty. Epistemic uncertainty is reducible and is more subjective in nature, while on the other hand aleatory uncertainty cannot be reduced and is more random in nature.

Researchers have been using different models in order to create a good model for uncertainty. The most acceptable models employed include Probability theory, possibility, fuzzy set and Dempster Shafer theory. Each of the model identified tries to incorporate uncertainty in the system by analyzing the level of information available and incorporating a mathematical representation of uncertainty in it. A certain model should not be used if its application requires higher information in order to successfully implement then the current ones available.

It is extremely difficult to understand the events and the likelihood of their occurrence. If an event is expected to occur multiple times then its frequency can be expressed in per day or per year or other similar terms. But if the event is expected to occur only once, then its frequency is expressed through probability. If an event it expected to occur frequently but it’s impossible or difficult to attach a numerical value to its occurrence then its frequency would be the probability of the frequency. The probabilistic risk assessment makes it compulsory to identify the information about all the occurrences. In case of the absence of the information the assumption that all the occurrences have equal likelihood should be used. This assumption is taken due to the axiom of addition that states that likelihood of all occurrences should be equal to 0. For example, if probability of event a happening is 0.25 then the probability of event a not occurring should be 0.75. In other words, a manager may have information regarding task duration which suggests that this task will take specific amount of time with probability of 0.75.This means that the uncertainty level is 0.25 (25 %). This is a direct result of such assumption.

This is a very rigid estimation regarding the likelihood of a threat materializing. There is a 25 % chance that the task duration will not be within the specified time duration which may leads to delays and sever management problems. In case of absence of information the probability should not be introduced because it could create a serious biasness in the model regarding the likelihood of its occurrence. Where the information is completely attainable and available in such scenarios probability theory is known to present the perfect results. Threat likelihood defines the probability of an event happening in the future based on the current or past circumstances. A higher degree of possibility or a lower degree of possibility does not necessarily imply that the event would have higher probability of occurrence or lower. But a high or low probability does define the possibility of an event occurring or not occurring.

Some threats would have a probability of zero attached to them but they should not be discounted because their occurrence could still materialize. On the other hand if an event has never happened that it is highly likely that it would not materialize in the future but that does not mean that it cannot occur. Fuzzy logic models have been extensively used in analyzing the information that is vague or ambiguous. But the fuzzy logic cannot deal with incomplete or inaccurate information. Ignorance cannot be analyzed by any means. Vagueness is defines as an uncertainty of a known event that is clearly measurable. To illustrate, any manager may assign a probability of zero for power blackout. Such blackout may dramatically extend and delay tasks duration. However, such event may happen and have devastating ramifications on project progress. To make sure that the probability of such event is zero, a manager should not only rely on occurrences. More information should be acquired from power supplier and should be handled using more sophisticated techniques (i.e. fuzzy logic) since pure statistical occurrences is not enough to model the uncertainty of such rare event. All of these different methods provide a great solution for incorporating different models of uncertainty that would further improve the efficiency and reliability of a model.

Traditionally project management was only concerned about the management of individual projects, but in the real world circumstances nearly all projects are carried out in sync or combination with other projects. Multiple Project Management is a new trend in the modern business community. Despite a large number of organizations working with multiple projects there is very little concrete data available regarding the effective management of such projects. Most projects follow the same cost and staffing requirements over the span of their life. Initially the projects tend to require lower budget and staff, then later on the cost and staff requirements start increasing drastically as the demand of the product increases in the market and drops with same speed in its final years. But, when it comes to the human resource and individual participants, their energy level, motivation and focus change over the life of the project. Using a common human resource on different projects increases the efficiency by eliminating idle time and the project also benefits from common expertise, improved communication and lower development costs.

It becomes extremely difficult to manage the allocation and consumption of resources in a multi-project environment. Resource management is the biggest issue that managers have to face in case of a multi-project management scenario. Human resource is one of the major issues that companies have to deal with in case of a project management; in addition to being the biggest cost in the project it is also the more complex. There are numerous methods described by management gurus, but very little is known about which application or method is more suitable in which conditions, therefore more work needs to be done in this area to determine the effective techniques of dealing with the allocation and management of human resource. The purpose of this thesis is to evaluate different methods of resource allocation in terms of their efficiency and application in different PM scenarios.

There are several jargons that are used interchangeable in the world of project management, since different countries and disciplines have different definition for these terms, we would be using the guidelines and definitions outlined by PMI standards to distinguish them. Portfolio refers to a collection of different programs, projects or other type of jobs merged or categorized together for the purpose effective management and ensuring coordination. A program is defined at a group of activities managed together in order to attain benefits and rewards in a coordinated fashion and these activities if managed individually would not results in a controlled or effective allocation of resources. In order to avoid confusion, we would be relying on the term of "Multi-project". Properly addressing uncertainty has a direct link to project portfolio management due to the fact that resources are shared among several projects at the same time. If a specific task in a specific project is delayed (or uncertainty regarding its duration was not taken into consideration), then resources which are used by this task will not be available for tasks of other projects. As a results, these other tasks will be delayed as well and sever domino effect of delays will be witnessed.

Resource allocation and management could have several objectives and goals, but in this thesis we would be focusing on attaining the tactical advantage for the business through effective resource allocation and scheduling. Tactical scheduling refers to the management of a project in such a way that the required activities could be performed on required time with the needed resources. When referring to resource, we mostly mean Human resource. Multi-project management scenarios with limited resources can be dealt with using different methods and techniques. The issue of Multi-project is dealt with by creating artificial project activity nodes at the beginning and end of multi as well as single projects. Therefore the multi-projects are treated in similar manner as a large project with multiple sub-projects or activities. Using this assumption it should be possible to resolve the issue of multi-project management using the individual PM methods such as Exact Method, Meta-heuristic and Heuristic Methods. Since the multi-projects have several activity nodes they are classified under a unique category called Network Optimization Algorithm. PM scenarios always carry a lot of embedded uncertainty in them. Algorithms to deal with uncertainty that are primarily used in single project management scenarios can be used as reference or beginning points for developing effective solutions for multi-project management scenarios. Some of the methods commonly used include Reactive Scheduling, Fuzzy Project Scheduling, Sensitivity Analysis and Stochastic Scheduling. Critical chain scheduling is a proactive approach towards dealing with Multi-project scheduling issues, its effectiveness primarily relies on the application of Theory of Constraint and Capacity constraining resources.

Many aspects of the project management rely heavily on the individual's ability, skills and prior knowledge. Different tasks have their own unique characteristics and require a set of different skills to efficiently deal with them. Several researches under the banner of Process Enactment are concerned in this very direction. This thesis propose a separate category called Resource modeling and Simulation to collect and allocate rigid resource definitions, policy driven resource allotment and continuous monitoring and simulation of the different states of resources. The multi-agent planning and resource scheduling methods are classified under a special category due to its distributed algorithm nature. According to this distributed class, resource scheduling for projects portfolio should not be global and central. Instead, each project will have its own scheduling agent to perform resources scheduling. The link among projects in the portfolio will be represented by uncertainty estimation. Each scheduler will generate scheduling decision for its project by taking uncertainty into consideration so that other projects can have the lowest uncertainty level when resources are distributed among projects.

Many of the activities in a project rely heavily on the people requiring proactive and instant responses for an effective Project management. Proactive methods of management place a vital importance on the application in real life situations when the project is continued with a continuous iteration. Proactive or agile practices can be extrapolated in order to deal with larger of Multi-project management scenarios. A sharing of human resources in terms of technicality and leadership can be used as a renewable resource from different teams or projects. Proactive approaches dictate that projects of different priorities should be dealt by the people of different competencies such as unattractive or difficult project should be handled by different teams in turns rather than attaching the entire management responsibility to a single individual.

Some managers tend to increase the size of a smaller project in order to incorporate multiple objective and products in it. The enlarged team could face the situation of too many cooks and compromise the effective management of the project. But in companies with smaller size or limited resources it can be very effective as a single team of programmers, network managers and business experts can deal with multiple smaller projects under the leadership of a single individual. Usually the members in a team are divided in 3 different groups, All-around members, service employees and individuals possessing expertise in a single field (Experts). Different project scheduling patterns and methods could be used to deal with different kinds of resources. Based on the aforementioned problems the author identified the following research gaps:

Most of existing research regarding uncertainty in scheduling problems follows an analytical approach to tackle this issue. Using mathematical approach can have many benefits to have a wide understanding of scheduling problems under uncertainty. However, it requires a lot of assumptions and simplifications to make such analysis feasible. There is a great need for tackling uncertainty issue from different perspective that the mathematical one.

Adoption of meta-heuristic techniques such as genetic algorithm to handle scheduling problems in project management can be easily found in literature. However, the common theme of such adoption is focused on the implementation aspects. In other words, answering how genetic algorithm can be used to perform scheduling is the main objective of many works in literature. While the implementation aspects are very important, the need to extend any of these implementations to handle complicated features of scheduling problem such as uncertainty is very important as well.

5.4 Methodology Strength

This research adopted Genetic Algorithm (GA) as the main methodology to find optimal scheduling decision since it is considered to be the most ideal approach for dealing with complex optimization problems. GA is primarily designed to replicate the methods used by the nature in evolution of the species. In nature each coming generation attempts to improve itself from the previous one by using the survival of fittest technique. GA method is also called population based method. As the search progresses under the GA method the system creates a group of population whereby they are improved with each successive search creating improved generations until the search is stopped or optimal solution is achieved.

When the GA is applied for scheduling problem, the system considers project tasks as a set of population, the system then attempts to improve the population throughout the search in order to create an optimal solution. Survival of the fittest technique also comes into the play whereby the system analyzes the parent population in order to create a list of task schedule that improve on the previous population. In GA method the task list is also referred to as chromosome, where each unique task is considered a strand of a gene. Diversification of possible scheduling solutions in the GA method is achieved by using the crossover method that identifies two or more crossover points in order to create a better sample. Intensification is incorporated in the model by using a different local search engine protocol that non-population based. If the search engine gets stuck in a non-optimal neighborhood the system employs the process called mutation in order to escape the non-optimal neighborhood. In mutation a single gene is chosen at random that is then switched with another gene as long as the switch does not compromise the feasible state of entire chromosome.

Under the GA method there are several different possibilities that are dependent upon the methods used to create new population sets. Mutation mechanism along with mutation strategy and parent chromosome heavily impacts the results of the method. GA method is considered to be very strong for global searches but is considered extremely weak in terms of local searches. GA has been found to produce the best results for resource scheduling problems, some researchers have also tried to refine GA method by adopting critical path concepts.

5.5 Uncertainties in Project Management

A wide range of uncertainties can be expected when managing any project. These uncertainties are usually dependent on project nature. Some projects may have different kinds of uncertainties compared to other projects. Examples of uncertainties that may face any project manager are:

- Time Uncertainty
- Cost Uncertainty
- Resources Uncertainty
- Regulations Uncertainty
- Weather Uncertainty
- Politics and Social Uncertainties

There are many other uncertainties that may affect managing process of any project. This thesis is dealing with resource scheduling problem. Hence, two of these uncertainties are of the most concern to this research.

5.5.1Time Uncertainty

Methods of traditional planning and their disability to handle unforeseen estimate that form the base for plans, and the inability to identify that work related to development does not experience progress in a linear way. In this regard, it must be noted, uncertain traditional critical path calculations give rise to comparatively smaller schedules than what should be expected in the realistic way. With the beginning point being a smaller schedule, delay is nothing but an obvious consequence. The next crisis (unforeseen delay event of any task in the project), presuming that progress of a task takes place at a consistent rate, stops project managers from sensing the signs of delay at the nascent stage until it's an un-repairable situation when there is no option left but to slash requirement features, compromise qualitatively or schedule the project on a later date.

To avoid such situations, uncertainty about tasks duration should be taken into consideration so that any delay events can be expected to some extent. Instead of developing several execution plans for the project, one may assigns uncertainty levels for every task in the project and perform scheduling. In essence, such approach can interpreted as developing unlimited number of execution plans where any task may be delayed within specified limits of task durations. At the same time, project progress should benefit from finishing tasks early. This can be done by instructing genetic algorithm to generate scheduling decision which the most robust.

5.5.2Resources Uncertainty

Any project even remotely associated with estimations involves uncertainly and project management is no exception. Resource constitutes the key to the entire chain of project scheduling and execution and its availability plays a pivotal role in smooth functioning of an industrial project delivery. Scarcity of resources is a massive impediment that disrupt the entire system and the whole organizational procedure needs to be regulated on several fronts to combat the scarcity caused due to unforeseen problems like environmental constraints, uncertainties related to sudden change in climate that adverse effect communication thus in turn jeopardizing access to resources and so on.

Under such circumstances, the only choice for the organization is to accept the crisis and adjust to the environment by incorporating few regulations in their organizational structure and behavior mainly personnel, procedural, process, and strategic. An organization is always vulnerable to fall prey to resource scarcity. This is a situation that can hit any organization at any point and thus uncertainty and risk is always present in every organization. Even in the most stable and efficient organizations that have concrete, stable and validated progress in their projects, they can run into problems with resource availability because unfortunate weather conditions, communication breakdown, natural calamities and similar difficulties can take place in case of any organization and this does cause urgent action to mitigate the problem with possibly the most effective solution.

As far as uncertainly related to resource availability is concerned, three important factors have to be considered. Firstly, one has to understand how critical the impact of lack of resources can be on the results of a project. Secondly, one has to realize the indispensability of gaining exhaustive knowledge about resources, their nature and availability for project managers since such insight will heavily benefit the organization during times of unforeseen crisis. Thirdly, since uncertainly is an integral part of project management as something can always runs a risk of occurrence, one must be aware of the advantages and as well as flipside of initiatives that might be successful in accommodating effective practices in order to minimize assurance with regard to resource availability with the help of proficient planning and tracking of projects.

As seen in previous chapter, analysis of resources availability shows that the proposed scheduling approach which takes uncertainty into consideration rarely lead to full consumption of resources. In other words, usually there are available resources that can be used in case of any unforeseen resources needs. The scheduling generated by the proposed approach doesn't try to schedule as many tasks as possible at the same time which may reduce resources availability to zero. Instead, scheduler assumes that tasks may be finished earlier due to uncertainty. Hence, the optimal approach is to try to increase resources availability as much as possible throughout project execution as seen in the analysis of resources availability.

5.6 Modeling Resource Scheduling

Resources scheduling is a very complex issue with many aspects that cannot be all considered. However, some of these aspects are more important than others in achieving an optimal scheduling. There are four main characteristics for successful optimization of resources scheduling. These include scheduling objective, network complexity, resource contention and resource distribution. Three scheduling objectives which are project lifespan minimization, average start time minimization and average finish time minimization were used. With regard to network complexity, resource contention and resource distribution, three projects of different sizes (30, 60 and 90 tasks) were evaluated. Each one of these projects has three different versions. The first version of the first project has the least network complexity, resource contention and resource distribution; while the third version of the third project has the highest network complexity, resource contention and resource distribution.

These four aspects are considered to be very important in proper resolution of projects in different situations. There are several different kinds of objective functions that managers rely upon for resources scheduling. The most popular measure is minimization of project duration. In addition to this measure there are other different kinds of objective functions used by the managers including total project delay, overall project cost and minimization of project cost. Studies have confirmed that the performance is heavily influenced by different objectives that are used.

Project networks with lesser complexity tend to be less constrained as compared to the alternatives. It is suggested that an adaptive number form should be used by the managers to normalize the complexity of the network. In a multi-project a composite measures should not be employed as there is very little evidence regarding their effectiveness of ineffectiveness in such situations. Three complexity issues may not be same as a one high or two lower complexity problems despite the fact that both provide similar averages. Therefore it is important to incorporate or maintain the distinctive features of the projects.

Researchers have developed several different measures to distribute the resources and determine their availability. The most commonly used measures include resource factor, a measure indicative of an average number of activities or resources utilized, and resource strength that expresses the requirement of the activities and availability of resource against such requirements. However these resource distribution measures may not be as effective in a Multi-project environment and require a distinct measure in resources scheduling. Average Resource Loading Factor seeks to identify whether the total resources of the project are saved at the front of back of the constraint chain. It tends to have several critical problems, first it can become biased and lose its ability to distinguish between different cases when percentages are used. Second is that despite its widespread application, it has been primarily created for single projects. A normalization approach can be used to counter these problems.

Utilization factor is used to deal with the resource contention issues. This factor is calculated by comparing the resource required with resource availability in a specific period. If the factor is estimated at less than 1 then it is assumed that there is no resource contention amongst the projects. An average factor is also used to decrease the computational intensity between different intervals.

5.7 Evaluation and Testing

The proposed solution is evaluated using simulation approach. The reason behind using this approach of testing is to investigate the abstract performance of the proposed genetic algorithm without the effects of other factors which are found in reality such as manager preferences (a manager may prefer to schedule a task on specific time because of reasons other than scheduling optimality). The performance of the proposed resources scheduling mechanisms has to be proven before they can be introduced in real project portfolio management environment. An actual implementation of the proposed resources scheduling mechanisms using Matlab environment was performed. Matlab is chosen because it is a very powerful developing environment for solution based on mathematical programming techniques. Also, it has very good support for artificial intelligence techniques as well. The support for mathematical programming and artificial intelligence techniques is provided in the form of toolboxes. By using these toolboxes, the researcher can reduce the coding time greatly. Every aspect of the proposed resources scheduling mechanism were traced and evaluated in very efficient way.

The simulation experiments were conducted using PSPLIB which is a well-known benchmark in resources scheduling literature. This benchmark includes set of projects where the each project has a set of tasks. The resource dependency for each task in the benchmark this predefined and fixed. Therefore, different resources scheduling mechanisms can be compared by equating their performances when they are used on the same set of projects in the benchmark. Using this benchmark allowed us to compare the proposed resources scheduling mechanisms with the previously published solutions in the literature.

A high-performance computation environment is constructed using cloud computation approach. The reason behind using cloud computation is the fact that the adopted search technique, which is genetic algorithm, requires a considerable computational resources and extensive execution time. Acquiring the necessary hardware to perform simulation experiments would be very expensive. A cheaper and more effective approach is to use cloud services. These services have been used by many scientists in many fields around the globe to perform their simulation experiments.

5.8 Research Findings

Any research effort will produce some finding which may have practical consequences. These practical consequences will greatly depend on findings quality and generalizability. The proposed scheduling solution was evaluated based on widely adopted data set of resource scheduling problem. In addition, the proposed scheduler was evaluated based on a real project. The main findings in this research are:

- Performed literature review suggests that achieving optimality in resource scheduling of project management can only be achieved by utilizing meta-heuristic or artificial intelligence techniques. These types of techniques will prohibited due to their expensive needs in term of computation and time. As a result, most work in literature avoided using these techniques. However, in the recent years computation capabilities of modern computers and the cost of hardware were reduced dramatically. This led to having more research works which utilize genetic algorithm and other materialistic approaches to show in literature in recent years. Work presented in this research contributes to the literature in this direction which is becoming more popular. Hence, future researchers may use this work to advance solutions for resources scheduling in project management. The adoption of findings in this research was made easier for future researchers since source code developed during investigation process was made available (check appendix). As a result, with regards to literature, this research has higher quality than many other works in literature since reducibility of findings can be much easier for outside researchers.

- Most implementations of genetic algorithm in literature to tackle resource scheduling problem in project management require a lot of modifications which negatively affect performance. At the same time, most of these modifications are unrealistic which lead to having unrealistic solutions for resources schedule. This problem is widespread in literature. Having unrealistic assumptions will reduce quality of finding greatly. In this research, the author tried to adopt the least amount of assumptions as best as possible. Hence, the proposed solution is more realistic than many other solutions which are existing in literature. This leads to having a higher-quality for the provided solution.
- The negative aspect of uncertainty can greatly reduce effectiveness of resource scheduling. As the complexity of a project increases, the negative impact of uncertainty increases exponentially. This was one of the main important finding in this research. Results showed that as uncertainty levels increases (increased from 5 % to 25 %), the project lifespan increases exponentially. This suggests that one of the most effective ways to shorten project lifespan is by reducing uncertainty regarding tasks durations as much as possible. In other words, as seen in previous chapter, projects with 25 % uncertainty have much longer lifespan than projects with lower uncertainty of 5 %. Reaching these findings was achieved by implementing the proposed resource scheduling approach using different projects with distinct sizes, complexities and resources availability. It was apparent that as the size of a project (which is the number of tasks) increases, the complexity of resources scheduling increases

as well. The same happens for tasks dependency and distribution of resources. Without taking uncertainty into consideration when developing resource scheduling solutions, quality and reliability of project management will be affected negatively. The proposed solution can lead to very reliable scheduling decisions which eliminate any expensive rescheduling process.

In addition, results of evaluations suggest that improvement of scheduling process can be achieved manipulating genetic algorithms parameters such as population size and maximum number of generations.

5.9 Revisiting Aim and Objectives

This research was focused on scheduling problem in project management under uncertainty. Four main objectives were set for this research which are:

- To tackle resource scheduling in project managements in a way that addresses shortcomings in literature especially uncertainty effect.
- To use advanced meta-heuristic approach such as genetic algorithm to build an automated resource scheduler under uncertainty.
- To evaluate and test proposed approach based on adopted standards in literature so that valid comparison can be established.
- To evaluate the proposed approach based on real resource scheduling task in real project management situation.

For the first objective, literature review showed that most existing studies try to tackle uncertainty issue by mathematically modeling it under unrealistic assumption. This research eliminates these assumptions which provide higher quality investigation of the problem. At the same time, better generalizability is achieved since the proposed solution is not restricted to specific situations under specific assumptions.

To achieve second objective, a new genetic coding approach is proposed so that uncertainty can incorporated in meta-heuristic optimization. It is well known that meta-heuristic optimization can find better solutions (scheduling decisions) than heuristic approaches. Having better scheduler certainly improves quality of the provided solution. Simultaneously, meta-heuristic optimization is general by its nature. Hence, the proposed scheduling approach has the highest level of generalizability in term of optimization.

Both of third and fourth objectives were achieved by conducting extensive evaluation on well-known benchmark and real project. Data set of projects used to perform evaluation is widely used in literature where all aspects of project structure and resources distribution are considered. Hence, evaluating the proposed solution using this data set guarantees both of quality and generalizability.

5.10 Implication and Applicability of Findings to PM

There are so many aspects of project management that need to be addressed properly to have an effective and efficient project execution. Scheduling is certainly one of the most important aspects. Having an optimal scheduling process is one the main criteria of successful projects. This research pointed out that uncertainty issue is very critical for effective scheduling process. From the conducted literature review, the uncertainty problem appears to be less important than other issue of scheduling such as the trade-off between cost and time. However, results in the previous chapter clearly suggest that uncertainty has dramatic impact on project lifespan which is the most important characteristics of any project. In other words, this research highlights that research on project management should provide more focus on uncertainty effect on scheduling process and project management in general. Similarly, the use of abstractive approach in designing the proposed solution allows its applicability to and scheduling setting in any project environment. This applicability was confirmed in previous chapter by scheduling a real project and achieving a better scheduling decision.

5.11 Research Contributions

This research proposed a scheduling solution to address uncertainty problem. The main contributions to literature are:

- Resources scheduling approach for project management which based on Genetic Algorithm. This approach incorporates the uncertainty in its operation. Most of the work in literature use genetic algorithm for resources scheduling without taking uncertainty into consideration. It worth mentioning that there are few works in literature which addresses uncertainty of resources scheduling based on analytical approach and mathematical modeling. However, the complexity of uncertainty issues in resources scheduling requires many assumptions to perform feasible mathematical modeling which would lead to unrealistic solutions. As a result, this research tries to use meta-heuristic technique which is genetic algorithm. Utilizing this technique doesn't require an overall analytical model of resources scheduling problem.
- An encoding technique to utilize binary Genetic Algorithm without extensive modifications as seen in literature. These modifications may negatively impact genetic algorithm implementation. As mentioned in the previous chapters,

genetic algorithm is designed to handle problems that can be included using binary encoding. However, resource scheduling problems are usually very hard to encode due to their dynamic nature. In other words, there is no universal structure for all projects with task dependency so that a universal encoding can be established. Each project is unique with its number of tasks, dependency and resources availability. Having an encoding technique that can be applied universally to all projects will be highly appreciated in literature. Nevertheless, the proposed encoding approach introduces redundancy in the encoding process. This mean that a project structure may have different codes that represent the same structure. This is a trade-off between generalizability and compactness.

- Finally, a simulation technique to simulate scheduling process computation in a way that allows utilization of meta-heuristic approaches was implemented. This simulator is developed using Matlab environment which is widely used among interested research in this type of research. Matlab code is very easy to understand compared to other programming language such as Java and C++. Developing the proposed simulator using Matlab will ease reproducibility of results generated in this research. At the same time, any interested researcher will have a much higher level of simplicity when implementing and developing their own solutions based on the work presented in this research.

In addition, a lot of related literature was reviewed which can be very useful for any interested researcher.

5.12 Research Limitations

The main and obvious limitation in this research is the inability to construct a data set of real projects. Such dataset would be very valuable to have an extensive evaluation of the proposed solution according to real projects. However, acquiring information about real project is very difficult due to many reasons such as week documentation and confidentiality regulations. As a result, only artificial dataset of projects was used in this research during simulation stage. Nevertheless, this dataset is widely used in literature. At the same time, having a common benchmark of projects would ease comparison among different solutions in literature.

5.13 Future Research

While genetic algorithm is natural choice for optimization of resource scheduling in project managements; other meta heuristic approach can be used as well. So far, works in literature are not able to modify these other approach in a way that achieves superiority. This effort can be very important contribution to literature. For example, one may use simulated annealing instead of genetic algorithm to find a better schedule solution. Other meta heuristic approaches can be used as well.

Another way to extend research in this thesis is by investigating other types of uncertainty. As mentioned before, there are many versions of uncertainty such as cost uncertainty and resources uncertainty. It will be very interesting to study how the proposed solution can be used to incorporate other types of uncertainty. Surely, some modification may be needed to extend the proposed solution in this thesis.

References

Gonçalves, J. F., Mendes, J. D. M., & Resende, M. G. (2008). A genetic algorithm for the resource constrained multi-project scheduling problem. European Journal of Operational Research, 189(3), 1171-1190.

A. Pritsker, L. Watters, and P. Wolfe. (1969). Multiproject scheduling with limited resources: A zero–one programming approach. Management Science, 16:93–107.

A. Schirmer and S. Riesenberg. (1998). Class-based control schemes for parameterized project scheduling heuristics. Manuskripte aus den Instituten fur Betriebswirtschaft- ¨ slehre 471, Universitat Kiel, Germany.

A. Schirmer and S. Riesenberg, (1997). Parameterized heuristics for project scheduling — Biased random sampling methods. Manuskripte aus den Instituten fur Betrieb- ¨ swirtschaftslehre 456, Universitat Kiel, Germany.

A. Schirmer. (2000). Case–based reasoning and improved adaptive search for project scheduling. Naval Research Logistics, 47:201–222.

A. Sprecher (2002). Network decomposition techniques for resource-constrained project scheduling. Journal of the Operational Research Society, 53(4):405–414.

A. Sprecher (2000). Scheduling resource-constrained projects competetively at modest memory requirements. Management Science, 46:710–723.

A. Thesen (1976). Heuristic scheduling of activities under resource and precedence restrictions. Management Science, 23:412–422.

Aaltonen, K. (2011). Project stakeholder analysis as an environmental interpretation process. International journal of project management, 29(2), 165-183.

Agarwal, A., Colak, S., & Erenguc, S. (2011). A neurogenetic approach for the resource-constrained project scheduling problem. Computers & Operations Research, 38(1), 44-50.

Artigues, C., Demassey, S., & Néron, E. (2008). Resource-constrained project scheduling models, algorithms, extensions and applications. London: Wiley.

B. Pollack-Johnson. (1995). Hybrid structures and improving forecasting and scheduling in project management. Journal of Operations Management , 12:101–117.

Babu, S., & Shah, M. (2013). Meta Heuristic Approach for Automatic Forecasting Model Selection. International Journal of Information Systems and Supply Chain Management (IJISSCM), 6(2), 1-16.

Bertsekas, D. P. (1999). Nonlinear programming. Athena Scientific Press.

Birbil, Ş. İ., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. Journal of global optimization, 25(3), 263-282.

Birge, J. R., & Louveaux, F. (2011). Introduction to stochastic programming. Springer.

Boyd, S. P., & Vandenberghe, L. (2004). Convex optimization. Cambridge university press.

Bullough, W. S. (2013). The evolution of differentiation. Elsevier.

C. Artigues, P. Michelon, and S. Reusser. (2003). Insertion techniques for statric and dynamic resource–constrained project scheduling. European Journal of Operational Research, 149:249–267.

C. E. Bell and J. Han. (1991). A new heuristic solution method in resource-constrained project scheduling. Naval Research Logistics, 38:315–331.

Cai, X., Wu, X., & Zhou, X. (2014). Optimal stochastic scheduling. New York: Springer.

Camila Costa Dutra, José Luis Duarte Ribeiro, Marly Monteiro de Carvalho, (2014), An economic–probabilistic model for project selection and prioritization, International Journal of Project Management, Available online 4 February 2014.

Caniels, M. C., & Bakens, R. J. (2015). Project Management Information Systems in a Multi-Project Environment. In Handbook on Project Management and Scheduling Vol. 2 (pp. 1355-1383). Springer International Publishing.

Chandrasekaran, M., & Muralidhar, M. (2013). Online optimization of multipass machining based cloud computing. International Journal of Advanced Manufacturing Technology, 65(1-4), 239-250.

Childe, S. (1997). An introduction to computer aided production management. London: Chapman and Hall.

D. Debels, B. De Reyck, R. Leus, and M. Vanhoucke. (2004). A hybrid scatter search / Electromagnetism meta–heuristic for project scheduling. European Journal of Operational Research.

D. F. Cooper. (1977). A note on serial and parallel heuristics for resource–constrained project scheduling. Foundations of Control Engineering , 2:131–133.

D. F. Cooper. (1976). Heuristics for scheduling resource–constrained projects: An experimental investigation. Management Science, 22:1186–1194.

D. Merkle, M. Middendorf, and H. Schmeck. (2002). Ant colony optimization for resourceconstrained project scheduling. IEEE Transactions on Evolutionary Computation, 6:333–346.

Dantzig, G. B. (1965). Linear programming and extensions. Princeton university press.

Dasgupta, D., & Michalewicz, Z. (2013). Evolutionary algorithms in engineering applications. Springer Science & Business Media.

Deblaere, F., Demeulemeester, E., & Herroelen, W. (2011). Proactive policies for the stochastic resource-constrained project scheduling problem. European Journal of Operational Research, 214(2), 308-316.

Detienne, B., Sadykov, R., & Tanaka, S. (2015). The two-machine flowshop total completion time problem: Branch-and-bound algorithms based on network-flow formulation. Retrieved from https://hal.inria.fr/hal-01248318/document.

Dowsland, K. A., & Thompson, J. M. (2012). Simulated annealing. In Handbook of Natural Computing (pp. 1623-1655). Springer Berlin Heidelberg.

E. A. Elsayed. (1982). Algorithms for project scheduling with resource constraints. International Journal of Production Research , 20:95–103.

E. Demeulemeester and W. Herroelen. (2002). Project scheduling – A research handbook . Kluwer Academic Publishers, Boston.

E. M. Davies. (1973). An experimental investigation of resource allocation in mulitactivity projects. Operations Research Quarterly , 24:587–591.

E. Pinson, C. Prins, and F. Rullier. (1994). Using tabu search for solving the resourceconstrained project scheduling problem. In Proceedings of the fourth international workshop on project management and scheduling, pages 102–106. Leuven, Belgium.

E. W. Davis and J. H. Patterson. (1975). A comparison of heuristic and optimum solutions in resource–constrained project scheduling. Management Science, 21:944–955.

Encarnacao, J. L., & Schhlechtendahl, E. (2012). Computer aided design: Fundamentals and system architectures. Berlin: Springer.

F. F. Boctor (1996). An adaptation of the simulated annealing algorithm for solving resource-constrained project scheduling problems. International Journal of Production Research, 34:2335–2351.

F. F. Boctor. (1990). Some efficient multi–heuristic procedures for resource–constrained project scheduling. European Journal of Operational Research , 49:3–13.

F. Glover, M. Laguna, and R. Marti. (2000). Fundamentals of scatter search and path relinking. Control and Cybernetics, 29(3):653–684.

G. E. Whitehouse and J. R. Brown. (1979). GENRES: An extension of Brooks algorithm for project scheduling with resource constraints. Computers & Industrial Engineering , 3:261–268.

G. Ulusoy and L. Ozdamar. (1989). Heuristic performance and network/resource charac- ¨ teristics in resource-constrained project scheduling. Journal of the Operational Research Society, 40:1145–1152.

Gido, J., & Clements, J. (2014). Successful project management. Cengage Learning.

Glover, F., & Laguna, M. (2013). Tabu Search∗ (pp. 3261-3362). Springer New York.

H. E. Mausser and S. R. Lawrence. (1995). Exploiting block structure to improve resource– constrained project schedules. Technical report, Graduate School of Business Administration, University of Colorado.

Hanisch, B., & Wald, A. (2011). A project management research framework integrating multiple theoretical perspectives and influencing factors. Project Management Journal, 42(3), 4-22.

He, L., & Zhang, L. (2013). Dynamic priority rule-based forward-backward heuristic algorithm for resource levelling problem in construction project. Journal of the Operational Research Society, 64(8), 1106-1117.

Heagney, J. (2012). Fundamentals of project management. AMACOM Div American Mgmt Assn.

Helton Cristiano Gomes, Francisco de Assis das Neves, Marcone Jamilson Freitas Souza, (2014), Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, Computers & Operations Research, Volume 44, April 2014, Pages 92-104.

Hendrickson, C., & Tung, A. (2008). Advanced scheduling techniques. Project Management for Construction. Retrieved from ttp://pmbook.ce.cmu.edu/11_Advanced_Scheduling_Techniques.html

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. Neural networks, 4(2), 251-257.

J. Alcaraz and C. Maroto. (2001). A robust genetic algorithm for resource allocation in project scheduling. Annals of Operations Research, 102:83–109.

J. Alcaraz, C. Maroto, and R. Ruiz. (2004). Improving the performance of genetic algorithms for the RCPS problem. Proceedings of the Ninth International Workshop on Project Management and Scheduling, pages 40–43.

Kaiafa, S., & Chassiakos, A. (2015). A genetic algorithm for optimal resource-driven project scheduling. Procedia Engineering, 123, 260-267.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications.Artificial Intelligence Review, 42(1), 21-57.

Kerzner, H. R. (2013). Project management: a systems approach to planning, scheduling, and controlling. John Wiley & Sons.

Kolisch, R. (1995). Project scheduling under resource constraints - efficeint heuristics for several problem classes. Berlin: Springer.

Kolisch, R. (2013). Project scheduling under resource constraints: efficient heuristics for several problem classes. Springer Science & Business Media.

Kolisch, R. and A. Sprecher (1996): PSPLIB - A project scheduling library, European Journal of Operational Research, Vol. 96, pp. 205--216.

Koné, O., Artigues, C., Lopez, P., & Mongeau, M. (2013). Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. Flexible Services and Manufacturing Journal, 25(1-2), 25-47.

L. Ozdamar and G. Ulusoy. (1994). A local constraint based analysis approach to project ¨ scheduling under general resource constraints. European Journal of Operational Research, 79:287–298.

L. R. Shaffer, J. B. Ritter, and W. L. Meyer. (1965). The critical-path method. McGraw-Hill, New York.

Laguna, M., & Marti, R. (2012). Scatter search: methodology and implementations in C (Vol. 24). Springer Science & Business Media.

Leach, L. P. (2014). Critical chain project management. Artech House.

Meredith, J. R., & Mantel Jr, S. J. (2011). Project management: a managerial approach. John Wiley & Sons.

Mohan, B. C., & Baskaran, R. (2012). A survey: Ant Colony Optimization based recent research and implementation on several engineering domain.Expert Systems with Applications, 39(4), 4618-4627.

Muhlenbein, H., Schomisch, M., & Born, J. (1991). The parallel genetic algorithm as function optimizer. Parallel computing, 17(6), 619-632.

Pinedo, M. (2012). Scheduling: Theory, algorithms, and systems. New York: Springer.

Project Management Institute. (2013). A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Project Management Institute, Incorporated.

R. Kolisch and A. Sprecher. (1996). PSPLIB — A project scheduling problem library. European Journal of Operational Research , 96:205–216.

Rossi, M. E., & Deutsch, C. V. (2014). Uncertainty and Risk. In Mineral Resource Estimation (pp. 209-221). Springer Netherlands.

Schwindt, C., & Zimmermann, J. (2015). Handbook on project management and scheduling. Berlin: Springer.

Slowinski, R., & Weglarz, J. (2013). Advances in project scheduling. Elsevier.

Spalek, S. (2012). The role of project management office in the multi-project environment. International Journal of Management and Enterprise Development, 12(2), 172-188.

Turner, J. R. (2014). The handbook of project-based management (Vol. 92). McGraw-hill.

V. Valls, F. Ballestin, and M. S. Quintanilla. (2003). A hybrid genetic algorithm for the RCPSP. Technical report, Department of Statistics and Operations Research, University of Valencia.

Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. European Journal of Operational Research, 235(1), 62-72.

Vanhoucke, M. (2012). Project management with dynamic scheduling. Spirnger Berling Heidelberg.

Vanhoucke, M. (2013). Project management with dynamic scheduling: Baseline scheduling, risk analysis and project control. Berlin: Springer.

W. Herroelen, E. Demeulemeester, and B. De Reyck. (1998). Resource–constrained project scheduling — A survey of recent developments. Computers & Operations Research , 25(4):279–302.

Weglarz, J. (2012). Project scheduling: recent models, algorithms and applications (Vol. 14). Springer Science & Business Media.

Wei-xin Wang, Xu Wang, Xian-long Ge, Lei Deng, (2014), Multi-objective optimization model for multi-project scheduling on critical chain, Advances in Engineering Software, Volume 68, February 2014, Pages 33-39.

Weske, M. (2012). Business process management: concepts, languages, architectures. Springer Science & Business Media.

Yaghootkar, K., & Gil, N. (2012). The effects of schedule-driven project management in multi-project environments. International Journal of Project Management, 30(1), 127-140.

Zamani, R. (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. European Journal of Operational Research, 229(2), 552-559.

Zhang, L., Wu, Q., Chen, C., & Yue, Y. (2012). Solution for the Nonlinear Multi-Objective Model in Construction Projects Using Improved Particle Swarm Optimization.

Zhang, Z., Long, K., Wang, J., & Dressler, F. (2014). On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches. Communications Surveys & Tutorials, IEEE,16(1), 513-537.

Blazewicz, J, Lenstra, J.K, and Kan, R(1982), Scheduling subjects to resource constraints: Classification and complexity, Holland, Holland: North Holland Publishing Company, retrieved from: http://www.sciencedirect.com/science/article/pii/0166218X83900124

Lewis, J(2011), Project Planning, Scheduling and Control, USA, USA: Mc Graw Hill, retrieved from: http://www.amazon.in/Project-Planning-Scheduling-Control-Hands/dp/0071746528

Baptiste,P and Timkovsky, V (2001),On Preemption Redundancy in Scheduling Unit Processing Time Jobs on Two Parallel Machine, Operations Research Letters, 28(5), 205-212

Abdeddaim, Y, and Maler, O(2002), Preemptive Job-shop Scheduling using Stopwatch Automata, TACAS '02 Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, 113-126

Kumar, V(1992), Algorithms for Constraint Satisfaction Problems: A Survey, AI Magazine, 13, 32-44

Applegate, D. & Cook, W. (1991),A Computational Study of the Job-Shop Scheduling Problem, ORSA Journal on Computing, 3, 149-156

Weglarz, J and Jozfowska, J(2006), Perspectives in Modern Project Scheduling, New York, NY: Springer

Mokhtari, H, Kazemzadeh, R.B, and Salmasnia, A(2011), Time Cost Trade Off Analysis in Project management: an Ant System Approach, IEEE Transactions on Engineering and Management, 58(1)

Lambersonm L.R. and Hocking, R. R(1970), Optimum time compression in project scheduling, Manage Sci, 16, 597-606

Vrat, P and Kriengkrairut,C(1986), Goal programming model for project crashing with piecewise linear time-cost trade-off, Eng. Costs Prod. Econ, 10, 161-172

Elbeltagi, E(2002), Time Cost Trade Off graphs retrieved from: http://osp.mans.edu.eg/elbeltagi/CM%20P%20Time-Cost.pdf

Klein, R(2000), scheduling of Resource Constrained Projects, New York, NY: Springer

Artigues, C, Demassey, S, and Neron, E(2008), Resource Constraint Project Scheduling: Models, Algorithms, Extensions and Applications, UK and USA, UK: ISTE ltd, NJ: John Wiley and Sons

Killian, J, Fiege, U and Uetz, M(1998), Zero Knowledge and the Chromatic Number, Journal of Computer and System Sciences - Eleventh annual conference on structure and complexity, 57(2), 187-199

Potts,C. N. and Wassenhove,V(1993), Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, Journal of the Operational Research Society, 43 , 395-406.

Monma, C. L, Potts, C. N, (1993) Analysis of heuristics for preemptive parallel machine scheduling with batch setup times, Journal of Operations Research, 41, 981-993.

Chen, B(1993), A better heuristic for preemptive parallel machine scheduling with batch setup times, SIAM Journal of Computation, 22, 1303-1318.

Zdrzalka, S(1994), Preemptive scheduling with release dates, delivery times and sequence independent setup times, European Journal of Operational Research, 76, 60-71.

Schuurman,P, Woeginger, G. J(1999), Preemptive scheduling with job dependent setup times, Proceeding of the 10th ACM-SIAM Symposium on Discrete Algorithms,759-767.

Liu, Z and Cheng, T. C. E (2002), Scheduling with job release dates, delivery times and preemption penalties, Information Processing Letter, 82, 107-111.

Julien, F. M, Magazine, M. J and Hall, N. G(1997), Generalized preemption models for single-machine dynamic scheduling problems, IIE Transactions 29 , 359-372.

Vanhoucke,M, Demeulemeester,E and Herroelen, W(2000a), An exact procedure for the resource constrained weighted earliness-tardiness project scheduling problem, Annals of Operations Research, 102 , 179-196.

Kaplan, L.A(1988), Resource constrained project scheduling with preemption of jobs, Michigan, Michigan: University of Michigan

Demeulemeester,E and Herroelen, W(1996), An efficient optimal solution procedure for the preemptive resource constrained project scheduling problem, European Journal of the Operational Research, 90 , 334- 348.

Besikci, U., Bilge, U, and Ulusoy, G. (2012), Resource dedication problem in a multi-project environment, To appear in Flexible Services and Manufacturing Journal

Vanhoucke.M (2012), Project Management with Dynamic Scheduling- Baselines scheduling, risk analysis and project control, New York, NY: Springer

Hanchate, D, Thorat, Y, and Ambole, R(2012), Review on Multimode Resource Constrained Project Scheduling Problem, International Journal of Computer Science & Engineering Technology, 3(5), 155-158

M¨ohring R. H. (1984), Minimizing costs of resource requirements in project networks subject to a fixed completion time,Operations Research, 32: 89- 120

Kr¨uger, D. and Scholl, A. (2009),A heuristic solution framework for the resource constrained(multi-)project scheduling problem with sequence dependent transfer times,European Journal of Operational Research, 197,492-508

Herreolen W, (2005). Project scheduling - theory and practice, Production and Operations Management, 14(4), 413-432.

Gademann, N. and Schutten, M. (2004), Linear programming based heuristics for project capacity planning,IIE Transactions, 37, 153-165.

Demeulemeester, E, (1995), Minimizing resource availability costs in time limited project networks, Management Science, 41(10), 1590-1598.

Besikci, U, Bilge and Ulusoy, G(2012), Resource dedication problem in a multi-project environment, To appear in Flexible Services and Manufacturing Journal

Herreolen, W and Demeulemeester, E(2002), Project Scheduling: A Research Handbook, New York, NY: Kluwer Academic publishers

Dasgupta, P.S. and Heal, G.M (1974), The Optimal Depletion of Exhaustible Resources, Review of Economic Studies, 3-28.

Dasgupta, P.S. and Heal, G.M(1979), Economic Theory and Exhaustible Resources, USA, USA: James Nisbet and Cambridge University Press.

Grossman, G.H. and Helpman, E(1991a), Quality Ladders in the Theory of Growth,Review of Economic Studies, , no(1)

Grossman, G.H. and Helpman, E(1991b),Quality Ladders and Product Cycles, Quarterly Journal of Economics, 106, no. 425, May.

Hartwick, J.M(1989). Non- renewable Resources: Extraction Programs and Markets, London, London: Harwood Academic.

Perez-Barahona , A(2006), Capital accumulation and exhaustible energy resources: a Special Functions case. CORE Discussion Papers. Forthcoming.

Perez-Barahona ,A. and Zou, B(2006a), A comparative study of energy saving technical progress in a vintage capital model, Resource and Energy Economics, 28, 181-191.

Perez-Barahona ,A. and Zou, B(2006b), Energy saving technological progress in a vintage capital model, Economic Modelling of Climate Change and Energy Policies. Edward Elgar, 166-179.

Smulders S, Nooij,M(2003), The impact of energy conservation on technology and economic growth, Resource and Energy Economics , 25, 59-79.

Solow, Robert M(1974), The Economics of Resources or the Resources of Economics, American Economic Review, American Economic Association, 64(2), 1-14.

Stiglitz J(1974), Growth with Exhaustible Natural Resources: Efficient and Optimal Growth Paths, Review of Economic Studies, 123-137.

Kavadias, S and Loch, C (2004), Project selection under Uncertainty, New York, NY: Springer

Miranda, E (2002), Planning and Executing Time Bound Projects, IEEE, 73-75

Koberg, C (1987), Resource Scarcity, Environmental Uncertainty, and Adaptive Organizational Behaviour, The Academy of Management Journals, 30(4), 798-807

Speirs, J, Glade, C and Slade, R (2015), Uncertainty in the availability of natural resources: fossil fuels, critical metals and biomass, Engineering and Physical Sciences Research Council.

Demeulemeester,E and Herroelen, W (2011), Robust Project Scheduling, USA, USA: Now Publishers Inc

Appendix A

Table 0.1: Resources dependency of first version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	4	0	0	0
3	4	10	0	0	0
4	6	0	0	0	3
5	3	3	0	0	0
6	8	0	0	0	8
7	5	4	0	0	0
8	9	0	1	0	0
9	2	6	0	0	0
10	7	0	0	0	1
11	9	0	5	0	0
12	2	0	7	0	0
13	6	4	0	0	0
14	3	0	8	0	0
15	9	3	0	0	0
16	10	0	0	0	5
17	6	0	0	0	8
18	5	0	0	0	7
19	3	0	1	0	0
20	7	0	10	0	0
21	2	0	0	0	6
22	7	2	0	0	0
23	2	3	0	0	0
24	3	0	9	0	0
25	3	4	0	0	0
26	7	0	0	4	0
27	8	0	0	0	7
28	3	0	8	0	0
29	7	0	7	0	0
30	2	0	7	0	0
31	2	0	0	2	0
32	0	0	0	0	0

Table A.0.2: Resources dependency of second version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	2	3	3	8
3	5	2	3	0	1
4	9	7	0	0	0
5	3	5	1	0	1
6	6	4	8	1	0
7	8	0	10	0	0
8	8	0	0	0	5
9	10	5	0	0	0
10	1	0	7	0	0
11	5	2	3	0	0
12	7	10	6	0	8
13	10	0	0	4	0
14	3	1	0	6	0
15	2	7	7	0	0
16	6	3	10	0	4
17	4	2	10	0	5
18	5	10	0	3	0
19	6	8	0	9	4
20	6	1	0	0	0
21	7	4	7	0	0
22	3	0	6	0	9
23	8	3	0	3	5
24	8	0	0	7	0
25	4	3	8	0	0
26	2	2	1	0	0
27	3	0	0	0	2
28	5	1	7	9	0
29	3	0	0	9	7
30	2	0	5	0	8
31	8	0	0	0	2
32	0	0	0	0	0

Table 0.3: Resources dependency of third version of project 1.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	7	3	10	4	4
3	10	5	8	9	8
4	2	4	4	3	9
5	6	7	5	9	10
6	8	4	7	3	4
7	1	2	6	9	7
8	8	3	7	9	8
9	1	10	4	8	3
10	7	10	10	3	1
11	4	4	1	4	5
12	6	2	6	6	2
13	4	7	7	8	10
14	9	7	9	8	8
15	10	8	1	3	9
16	1	4	6	2	7
17	7	10	2	8	5
18	8	6	3	3	8
19	6	7	9	1	5
20	2	6	2	1	2
21	5	6	5	9	1
22	6	2	3	9	7
23	2	2	10	8	4
24	2	10	10	7	7
25	6	6	7	9	5
26	9	8	6	4	7
27	4	7	5	7	7
28	10	2	10	9	4
29	3	7	9	6	6
30	5	10	4	6	4
31	8	9	4	4	9
32	0	0	0	0	0

Table A.0.4: Resources dependency for first version of second project.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	10	0	0	0
3	1	0	1	0	0
4	10	0	9	0	0
5	6	0	4	0	0
6	5	0	0	0	1
7	8	10	0	0	0
8	9	0	0	6	0
9	1	0	0	0	8
10	9	0	6	0	0
11	8	0	0	0	3
12	3	0	7	0	0
13	6	8	0	0	0
14	2	0	0	0	1
15	5	0	0	0	9
16	1	6	0	0	0
17	3	2	0	0	0
18	10	0	0	2	0
19	9	7	0	0	0
20	1	5	0	0	0
21	3	0	0	8	0
22	6	0	4	0	0
23	3	0	0	4	0
24	3	0	5	0	0
25	7	0	0	1	0
26	6	9	0	0	0
27	10	0	7	0	0
28	9	3	0	0	0
29	8	0	0	3	0
30	4	0	0	7	0
31	3	6	0	0	0
32	3	0	0	0	4
33	6	0	7	0	0
34	1	0	0	0	4
35	9	0	0	1	0
36	9	9	0	0	0
37	1	0	7	0	0
38	2	5	0	0	0
39	4	0	0	1	0
40	9	0	0	0	5
41	10	0	0	0	1
42	8	0	0	0	9
43	4	0	0	6	0
44	3	0	0	0	1
45	6	0	0	0	9
46	6	0	0	0	7

47	7	4	0	0	0
48	3	0	8	0	0
49	2	0	2	0	0
50	10	0	0	7	0
51	4	0	5	0	0
52	2	2	0	0	0
53	1	0	1	0	0
54	4	0	0	6	0
55	10	0	0	0	7
56	8	0	0	3	0
57	6	0	4	0	0
58	10	0	0	9	0
59	3	0	0	0	7
60	10	0	3	0	0
61	1	0	0	0	1
62	0	0	0	0	0

Table A.0.5: Resources dependency for second version of second project.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	2	3	10	5
3	2	6	7	6	0
4	3	1	0	9	0
5	5	1	3	2	0
6	1	0	0	5	0
7	5	0	0	0	5
8	9	0	0	0	2
9	9	0	10	1	5
10	5	0	0	0	1
11	6	3	10	0	2
12	9	0	0	0	5
13	1	0	0	6	3
14	6	5	0	0	0
15	4	9	0	4	8
16	10	0	10	2	0
17	6	0	0	7	0
18	4	0	0	2	5
19	8	6	0	2	0
20	7	3	0	0	1
21	4	0	0	0	10
22	3	9	0	7	9
23	3	8	1	5	0
24	1	0	8	4	0
25	9	0	0	10	0
26	10	0	0	2	7
27	6	0	10	5	0
28	3	7	8	0	0
29	10	0	0	9	7
30	9	5	1	0	0
31	4	8	0	0	0
32	1	0	6	0	0
33	1	0	0	5	9
34	10	6	7	6	5
35	5	0	5	0	7
36	8	8	6	6	5
37	6	3	0	0	5
38	3	10	9	9	0
39	8	6	10	6	0
40	2	6	4	8	0
41	2	5	10	0	3
42	8	0	5	0	0
43	1	3	0	0	0
44	8	3	4	0	0
45	8	0	0	4	9
46	4	0	0	0	6

47	5	3	7	0	0
48	4	5	1	0	0
49	4	0	5	0	0
50	6	8	0	8	6
51	9	3	0	0	0
52	8	8	0	10	0
53	3	0	2	0	0
54	1	3	3	0	7
55	6	0	0	2	6
56	7	0	0	7	0
57	2	3	0	0	2
58	4	3	2	0	9
59	5	0	0	5	0
60	1	4	0	9	0
61	6	7	1	0	0
62	0	0	0	0	0

Table 0.6: Resources dependency for third version of second project.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	1	3	5	3	1
3	9	5	1	9	1
4	3	2	2	1	9
5	6	1	1	2	6
6	7	4	10	10	10
7	4	4	2	5	9
8	1	5	4	1	9
9	3	4	8	5	6
10	4	8	3	9	5
11	2	6	3	10	1
12	10	8	2	2	1
13	2	4	4	2	8
14	1	7	8	3	2
15	5	6	2	7	10
16	4	3	8	3	2
17	4	10	9	10	9
18	10	2	3	10	8
19	9	1	3	10	10
20	8	1	8	5	5
21	2	10	8	8	9
22	10	1	6	3	5
23	7	8	3	2	9
24	4	9	7	1	5
25	1	1	8	6	4
26	3	7	3	1	7
27	8	9	5	3	9
28	4	5	6	2	6
29	2	1	4	3	4
30	3	1	1	1	9
31	6	8	1	10	3
32	3	10	2	5	1
33	4	2	10	8	4
34	10	4	3	6	10
35	3	9	6	9	9
36	4	4	9	6	6
37	10	6	2	7	5
38	3	3	10	10	1
39	5	6	10	5	2
40	6	7	4	7	3
41	7	7	10	9	6
42	9	4	1	5	6
43	3	4	6	1	3
44	2	10	3	6	5
45	2	7	4	4	10
46	4	4	10	8	9

47	8	9	1	4	6
48	4	10	5	7	6
49	4	10	6	7	2
50	2	9	4	5	4
51	2	10	5	8	1
52	9	1	7	8	4
53	8	10	5	1	2
54	2	1	9	8	8
55	10	3	7	2	10
56	4	10	10	1	6
57	10	10	3	4	4
58	10	10	2	3	7
59	8	6	5	7	6
60	4	5	5	1	5
61	8	7	8	9	1
62	0	0	0	0	0

Table 0.7: Resources dependency for first version of Project 3.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	8	0	0	0	3
3	10	0	0	7	0
4	1	2	0	0	0
5	2	0	0	0	7
6	10	0	0	10	0
7	8	0	0	0	6
8	6	8	0	0	0
9	9	0	0	0	4
10	1	0	2	0	0
11	8	0	7	0	0
12	2	0	0	6	0
13	7	0	0	0	10
14	2	0	0	0	10
15	7	8	0	0	0
16	4	0	2	0	0
17	10	0	0	0	6
18	9	2	0	0	0
19	9	0	0	6	0
20	3	0	7	0	0
21	7	0	6	0	0
22	2	0	0	0	6
23	9	9	0	0	0
24	6	2	0	0	0
25	3	1	0	0	0
26	3	0	0	10	0
27	6	0	0	9	0
28	5	0	0	7	0
29	6	0	0	9	0
30	8	7	0	0	0
31	5	4	0	0	0
32	4	0	0	1	0
33	7	0	7	0	0
34	7	0	0	0	1
35	2	8	0	0	0
36	8	0	4	0	0
37	4	0	9	0	0
38	9	0	6	0	0
39	5	0	0	8	0
40	7	0	0	0	7
41	6	0	0	0	5
42	8	0	0	9	0
43	8	4	0	0	0
44	1	0	0	4	0
45	1	0	0	9	0
46	8	0	0	4	0

47	6	0	4	0	0
48	9	0	0	6	0
49	1	0	0	0	5
50	8	0	0	0	3
51	8	0	0	7	0
52	10	6	0	0	0
53	3	0	0	5	0
54	6	0	0	0	5
55	3	0	8	0	0
56	10	0	0	6	0
57	5	0	0	7	0
58	2	0	0	7	0
59	3	5	0	0	0
60	3	0	4	0	0
61	3	0	5	0	0
62	5	0	7	0	0
63	9	0	3	0	0
64	5	3	0	0	0
65	4	0	1	0	0
66	8	0	0	0	2
67	9	0	0	0	1
68	4	0	8	0	0
69	2	8	0	0	0
70	7	0	4	0	0
71	2	0	0	0	3
72	1	0	10	0	0
73	5	7	0	0	0
74	9	0	0	0	5
75	4	0	0	5	0
76	2	0	0	0	2
77	7	0	8	0	0
78	8	0	2	0	0
79	8	0	0	0	2
80	1	0	0	0	4
81	6	0	0	10	0
82	10	0	0	5	0
83	9	0	0	0	9
84	2	7	0	0	0
85	7	0	0	0	5
86	9	0	0	3	0
87	8	0	10	0	0
88	1	0	8	0	0
89	2	3	0	0	0
90	9	0	6	0	0
91	3	0	6	0	0
92	0	0	0	0	0

Table A.8: Resources dependency for second version of Project 3.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	9	5	3	0	0
3	2	0	10	0	1
4	6	1	1	0	0
5	8	0	6	2	0
6	1	6	0	0	5
7	3	0	0	9	0
8	1	0	0	0	9
9	6	0	7	5	2
10	8	0	6	0	0
11	6	0	8	10	0
12	4	10	0	0	7
13	7	0	4	0	6
14	8	8	0	0	0
15	4	0	0	5	5
16	6	4	7	0	0
17	8	6	4	0	0
18	10	5	10	4	0
19	6	8	0	9	5
20	4	5	10	0	0
21	7	8	4	0	0
22	1	10	0	0	1
23	2	4	0	8	3
24	8	0	0	0	6
25	3	0	0	0	7
26	1	0	8	0	3
27	1	4	9	2	0
28	8	4	0	5	4
29	9	0	9	0	4
30	10	1	0	6	0
31	3	6	0	0	0
32	4	8	7	0	3
33	7	0	0	9	0
34	1	1	1	4	3
35	9	8	0	0	10
36	8	0	4	10	9
37	1	0	7	0	0
38	4	5	10	8	0
39	2	9	9	0	0
40	9	5	0	0	2
41	3	0	0	1	9
42	2	0	1	0	4
43	7	0	10	0	10
44	1	0	0	0	1
45	3	7	0	5	0
46	2	4	9	3	0

47	8	7	2	0	0
48	5	0	0	7	0
49	6	8	5	8	6
50	3	5	0	0	0
51	6	0	4	1	8
52	1	2	9	0	0
53	3	4	8	0	0
54	9	0	10	10	4
55	1	0	6	2	0
56	8	10	4	0	0
57	2	0	4	0	0
58	9	0	6	0	1
59	7	4	0	0	0
60	7	7	0	0	8
61	10	0	0	0	7
62	3	3	0	3	1
63	2	6	8	8	0
64	5	2	9	4	0
65	8	9	0	1	7
66	5	0	10	0	9
67	8	0	0	10	6
68	2	0	0	6	2
69	4	0	1	10	0
70	1	0	0	5	2
71	10	8	0	0	10
72	5	0	7	0	0
73	3	0	0	1	0
74	7	0	0	8	10
75	7	0	2	3	0
76	7	5	0	7	0
77	10	0	6	6	0
78	10	6	0	0	8
79	3	0	4	6	0
80	8	0	10	0	0
81	1	0	5	0	6
82	7	2	0	9	5
83	5	4	4	1	0
84	8	0	7	0	0
85	5	0	8	0	0
86	2	0	7	0	7
87	6	0	0	0	3
88	8	3	0	8	0
89	9	7	7	0	7
90	5	0	0	3	0
91	3	7	0	0	8
92	0	0	0	0	0

Table A.9: Resources dependency for third version of Project 3.

Tasks	Duration	Resource 1	Resource 2	Resource 3	Resource 4
1	0	0	0	0	0
2	2	7	10	7	9
3	7	8	3	9	1
4	7	3	8	6	1
5	10	7	2	2	8
6	8	3	1	2	7
7	6	8	1	9	3
8	9	9	9	5	2
9	3	6	8	2	3
10	7	1	1	4	9
11	3	3	1	3	6
12	3	6	8	3	8
13	2	8	7	3	1
14	4	6	3	6	8
15	10	10	1	9	3
16	10	1	6	9	9
17	10	6	4	8	4
18	8	4	3	8	7
19	1	8	3	2	1
20	6	2	10	10	4
21	4	10	8	9	2
22	9	6	3	10	4
23	3	4	4	8	9
24	10	9	2	3	7
25	4	3	7	2	8
26	1	7	1	8	9
27	1	7	6	5	8
28	8	1	4	9	7
29	6	8	4	6	4
30	4	4	2	1	2
31	5	10	7	3	6
32	9	10	6	3	3
33	6	2	2	8	6
34	5	9	6	4	10
35	8	3	7	7	3
36	9	9	9	1	7
37	10	10	3	5	4
38	4	1	1	10	5
39	3	8	2	2	8
40	5	9	8	8	1
41	8	6	1	10	5
42	4	10	2	2	6
43	1	1	10	9	5
44	9	10	6	8	9
45	9	5	10	3	7
46	6	4	9	5	3

47	4	5	8	1	5
48	9	6	4	2	7
49	2	7	1	4	6
50	3	1	1	2	1
51	4	6	8	4	2
52	10	5	10	1	3
53	6	8	2	7	5
54	3	5	1	8	8
55	7	1	3	2	9
56	6	7	5	9	6
57	3	9	9	3	9
58	2	3	2	1	10
59	9	3	7	7	6
60	9	7	8	7	1
61	9	5	1	2	3
62	3	2	6	4	1
63	9	4	7	9	1
64	10	3	2	4	8
65	5	5	9	7	8
66	2	9	3	9	7
67	7	9	2	8	4
68	6	1	7	10	9
69	9	3	3	10	9
70	5	6	5	8	2
71	8	2	4	3	2
72	2	2	5	6	5
73	2	5	1	1	8
74	8	9	5	3	7
75	10	6	3	8	7
76	4	3	9	10	8
77	1	5	3	9	1
78	2	3	5	10	5
79	1	7	7	5	7
80	4	2	8	2	1
81	10	10	9	2	1
82	1	10	4	8	9
83	4	10	7	8	7
84	2	10	2	2	3
85	8	9	6	8	10
86	4	1	10	6	4
87	8	9	2	1	5
88	3	2	6	4	5
89	4	3	3	8	10
90	10	2	7	3	9
91	2	1	9	9	5
92	0	0	0	0	0

Table 0.8: The uncertainty analysis for first project.

Task	Mean	Standard Deviation	Scale	Shape	Rate 1	Rate 2	Rate 3	Rate 4
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	7.67	0.58	7.87	23.91	23.0	33.22	17.89	30.67
3	6.33	3.21	7.17	2.63	35.89	23.22	19.0	19.0
4	5.67	3.51	6.41	2.1	20.78	7.56	5.67	22.67
5	4.0	1.73	4.49	3.05	20.0	8.0	12.0	14.67
6	7.33	1.15	7.73	11.1	19.56	36.67	9.78	29.33
7	4.67	3.51	5.16	1.55	9.33	24.89	14.0	10.89
8	8.33	0.58	8.57	17.97	8.33	22.22	25.0	36.11
9	4.33	4.93	4.47	1.08	30.33	5.78	11.56	4.33
10	5.0	3.46	5.54	1.64	16.67	28.33	5.0	3.33
11	6.0	2.65	6.75	3.02	12.0	18.0	8.0	10.0
12	5.0	2.65	5.64	2.62	20.0	31.67	10.0	16.67
13	6.67	3.06	7.51	2.94	24.44	15.56	26.67	22.22
14	5.0	3.46	5.69	1.93	13.33	28.33	23.33	13.33
15	7.0	4.36	7.87	2.03	42.0	18.67	7.0	21.0
16	5.67	4.51	6.18	1.4	13.22	30.22	3.78	30.22
17	5.67	1.53	6.16	5.64	22.67	22.67	15.11	34.0
18	6.0	1.73	6.58	4.5	32.0	6.0	12.0	30.0
19	5.0	1.73	5.52	4.61	25.0	16.67	16.67	15.0
20	5.0	2.65	5.64	2.62	11.67	20.0	1.67	3.33
21	4.67	2.52	5.28	2.53	15.56	18.67	14.0	10.89
22	5.33	2.08	5.94	3.85	7.11	16.0	16.0	28.44
23	4.0	3.46	4.49	1.53	10.67	13.33	14.67	12.0
24	4.33	3.21	4.92	1.78	14.44	27.44	20.22	10.11
25	4.33	1.53	4.81	3.83	18.78	21.67	13.0	7.22
26	6.0	3.61	6.77	2.17	20.0	14.0	16.0	14.0
27	5.0	2.65	5.67	2.52	11.67	8.33	11.67	26.67
28	6.0	3.61	6.82	2.22	6.0	50.0	36.0	8.0
29	4.33	2.31	4.92	2.5	10.11	23.11	21.67	18.78
30	3.0	1.73	3.41	2.31	10.0	16.0	6.0	12.0
31	6.0	3.46	6.77	2.3	18.0	8.0	12.0	22.0
32	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A.11: The uncertainty analysis for second project.

Task	Mean	Standard Deviation	Scale	Shape	Rate 1	Rate 2	Rate 3	Rate 4
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5.67	4.04	6.23	1.54	28.33	15.11	24.56	11.33
3	4.0	4.36	4.21	1.14	14.67	12.0	20.0	1.33
4	5.33	4.04	6.05	1.76	5.33	19.56	17.78	16.0
5	5.67	0.58	5.87	17.51	3.78	15.11	7.56	11.33
6	4.33	3.06	4.83	1.68	5.78	14.44	21.67	15.89
7	5.67	2.08	6.3	3.63	26.44	3.78	9.44	26.44
8	6.33	4.62	6.9	1.45	10.56	8.44	14.78	23.22
9	4.33	4.16	4.68	1.27	5.78	26.0	8.67	27.44
10	6.0	2.65	6.75	3.02	16.0	18.0	18.0	12.0
11	5.33	3.06	6.03	2.34	16.0	23.11	17.78	10.67
12	7.33	3.79	8.27	2.73	19.56	22.0	4.89	14.67
13	3.0	2.65	3.34	1.46	12.0	4.0	8.0	11.0
14	3.0	2.65	3.34	1.46	12.0	8.0	3.0	3.0
15	4.67	0.58	4.87	14.31	23.33	3.11	17.11	42.0
16	5.0	4.58	5.4	1.29	15.0	30.0	8.33	3.33
17	4.33	1.53	4.81	3.83	17.33	13.0	24.56	13.0
18	8.0	3.46	8.95	3.48	5.33	8.0	37.33	34.67
19	8.67	0.58	8.87	27.11	40.44	8.67	34.67	28.89
20	5.33	3.79	5.88	1.57	16.0	14.22	8.89	10.67
21	3.0	1.0	3.31	4.23	10.0	8.0	16.0	19.0
22	6.33	3.51	7.18	2.42	21.11	21.11	21.11	29.56
23	4.33	2.31	4.92	2.5	23.11	5.78	15.89	13.0
24	2.67	1.53	3.01	2.34	8.0	17.78	4.44	4.44
25	5.67	4.16	6.21	1.5	1.89	15.11	32.11	7.56
26	6.33	3.51	7.18	2.42	33.78	6.33	6.33	29.56
27	8.0	2.0	8.67	5.67	24.0	58.67	21.33	24.0
28	5.33	3.21	6.07	2.21	26.67	24.89	3.56	10.67
29	6.67	4.16	7.51	2.04	2.22	8.89	33.33	24.44
30	5.33	3.21	6.07	2.21	10.67	3.56	14.22	16.0
31	4.33	1.53	4.81	3.83	31.78	1.44	14.44	4.33
32	2.33	1.15	2.63	2.91	7.78	6.22	3.89	3.89
33	3.67	2.52	4.11	1.8	2.44	20.78	15.89	15.89
34	7.0	5.2	7.57	1.39	23.33	23.33	28.0	44.33
35	5.67	3.06	6.42	2.49	17.0	20.78	18.89	30.22
36	7.0	2.65	7.77	4.05	49.0	35.0	28.0	25.67
37	5.67	4.51	6.18	1.4	17.0	17.0	13.22	18.89
38	2.67	0.58	2.86	7.87	16.0	16.89	16.89	0.89
39	5.67	2.08	6.3	3.63	22.67	37.78	22.67	3.78
40	5.67	3.51	6.41	2.1	24.56	15.11	28.33	15.11
41	6.33	4.04	7.14	2.0	25.33	42.22	19.0	21.11
42	8.33	0.58	8.57	17.97	11.11	16.67	13.89	41.67
43	2.67	1.53	3.01	2.34	6.22	5.33	6.22	2.67
44	4.33	3.21	4.92	1.78	18.78	10.11	8.67	8.67

45	5.33	3.06	6.03	2.34	12.44	7.11	14.22	49.78
46	4.67	1.15	5.07	5.22	6.22	15.56	12.44	34.22
47	6.67	1.53	7.18	6.69	35.56	17.78	8.89	13.33
48	3.67	0.58	3.86	11.1	18.33	17.11	8.56	7.33
49	3.33	1.15	3.68	4.61	11.11	14.44	7.78	2.22
50	6.0	4.0	6.78	1.92	34.0	8.0	40.0	20.0
51	5.0	3.61	5.66	1.81	21.67	16.67	13.33	1.67
52	6.33	3.79	7.14	2.18	23.22	14.78	38.0	8.44
53	4.0	3.61	4.38	1.36	13.33	10.67	1.33	2.67
54	2.33	1.53	2.65	2.01	3.11	9.33	10.89	11.67
55	8.67	2.31	9.4	6.25	8.67	20.22	11.56	66.44
56	6.33	2.08	6.97	4.69	21.11	21.11	23.22	12.67
57	6.0	4.0	6.78	1.92	26.0	14.0	8.0	12.0
58	8.0	3.46	8.95	3.48	34.67	10.67	32.0	42.67
59	5.33	2.52	6.01	2.88	10.67	8.89	21.33	23.11
60	5.0	4.58	5.4	1.29	15.0	13.33	16.67	8.33
61	5.0	3.61	5.53	1.58	23.33	15.0	15.0	3.33
62	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A.12: The uncertainty analysis for third project.

Task	Mean	Standard Deviation	Scale	Shape	Rate 1	Rate 2	Rate 3	Rate 4
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	6.33	3.79	7.14	2.18	25.33	27.44	14.78	25.33
3	6.33	4.04	7.14	2.0	16.89	27.44	33.78	4.22
4	4.67	3.21	5.18	1.68	9.33	14.0	9.33	1.56
5	6.67	4.16	7.51	2.04	15.56	17.78	8.89	33.33
6	6.33	4.73	6.89	1.43	19.0	2.11	25.33	25.33
7	5.67	2.52	6.36	3.19	15.11	1.89	34.0	17.0
8	5.33	4.04	5.86	1.48	30.22	16.0	8.89	19.56
9	6.0	3.0	6.78	2.74	12.0	30.0	14.0	18.0
10	5.33	3.79	5.88	1.57	1.78	16.0	7.11	16.0
11	5.67	2.52	6.36	3.19	5.67	30.22	24.56	11.33
12	3.0	1.0	3.31	4.23	16.0	8.0	9.0	15.0
13	5.33	2.89	6.02	2.55	14.22	19.56	5.33	30.22
14	4.67	3.06	5.3	2.01	21.78	4.67	9.33	28.0
15	7.0	3.0	7.85	3.25	42.0	2.33	32.67	18.67
16	6.67	3.06	7.51	2.94	11.11	33.33	20.0	20.0
17	9.33	1.15	9.73	14.31	37.33	24.89	24.89	31.11
18	9.0	1.0	9.37	12.71	33.0	39.0	36.0	21.0
19	5.33	4.04	5.86	1.48	28.44	5.33	30.22	10.67
20	4.33	1.53	4.81	3.83	10.11	39.0	14.44	5.78
21	6.0	1.73	6.54	5.71	36.0	36.0	18.0	4.0
22	4.0	4.36	4.21	1.14	21.33	4.0	13.33	14.67
23	4.67	3.79	5.26	1.62	26.44	6.22	24.89	18.67
24	8.0	2.0	8.67	5.67	29.33	5.33	8.0	34.67
25	3.33	0.58	3.55	7.36	4.44	7.78	2.22	16.67
26	1.67	1.15	1.9	1.93	3.89	5.0	10.0	6.67
27	2.67	2.89	2.84	1.18	9.78	13.33	14.22	7.11
28	7.0	1.73	7.56	6.79	11.67	9.33	49.0	25.67
29	7.0	1.73	7.6	5.22	18.67	30.33	35.0	18.67
30	7.33	3.06	8.2	3.49	29.33	4.89	17.11	4.89
31	4.33	1.15	4.7	6.25	28.89	10.11	4.33	8.67
32	5.67	2.89	6.42	2.61	34.0	24.56	7.56	11.33
33	6.67	0.58	6.87	20.71	4.44	20.0	37.78	13.33
34	4.33	3.06	4.83	1.68	14.44	10.11	11.56	20.22
35	6.33	3.79	7.14	2.18	40.11	14.78	14.78	27.44
36	8.33	0.58	8.57	17.97	25.0	47.22	30.56	44.44
37	5.0	4.58	5.4	1.29	16.67	31.67	8.33	6.67
38	5.67	2.89	6.42	2.61	11.33	32.11	34.0	9.44
39	3.33	1.53	3.75	2.94	18.89	12.22	11.11	8.89
40	7.0	2.0	7.65	4.95	32.67	18.67	18.67	23.33
41	5.67	2.52	6.36	3.19	11.33	1.89	20.78	35.89
42	4.67	3.06	5.3	2.01	15.56	4.67	17.11	15.56
43	5.33	3.79	5.88	1.57	8.89	35.56	16.0	26.67
44	3.67	4.62	3.6	0.96	12.22	7.33	14.67	12.22

45	4.33	4.16	4.68	1.27	17.33	14.44	24.56	10.11
46	5.33	3.06	6.03	2.34	14.22	32.0	21.33	5.33
47	6.0	2.0	6.62	4.23	24.0	28.0	2.0	10.0
48	7.67	2.31	8.38	5.43	15.33	10.22	38.33	17.89
49	3.0	2.65	3.34	1.46	15.0	6.0	12.0	17.0
50	4.67	2.89	5.31	2.16	9.33	1.56	3.11	6.22
51	6.0	2.0	6.62	4.23	12.0	24.0	24.0	20.0
52	7.0	5.2	7.57	1.39	30.33	44.33	2.33	7.0
53	4.0	1.73	4.49	3.05	16.0	13.33	16.0	6.67
54	6.0	3.0	6.78	2.74	10.0	22.0	36.0	34.0
55	3.67	3.06	4.07	1.49	1.22	20.78	4.89	11.0
56	8.0	2.0	8.67	5.67	45.33	24.0	40.0	16.0
57	3.33	1.53	3.75	2.94	10.0	14.44	11.11	10.0
58	4.33	4.04	4.81	1.41	4.33	11.56	11.56	15.89
59	6.33	3.06	7.13	2.93	25.33	14.78	14.78	12.67
60	6.33	3.06	7.13	2.93	29.56	25.33	14.78	19.0
61	7.33	3.79	8.27	2.73	12.22	14.67	4.89	24.44
62	3.67	1.15	4.04	4.14	6.11	15.89	8.56	2.44
63	6.67	4.04	7.51	2.12	22.22	40.0	37.78	2.22
64	6.67	2.89	7.49	3.05	17.78	24.44	17.78	17.78
65	5.67	2.08	6.3	3.63	26.44	18.89	15.11	28.33
66	5.0	3.0	5.67	2.21	15.0	21.67	15.0	30.0
67	8.0	1.0	8.37	11.31	24.0	5.33	48.0	29.33
68	4.0	2.0	4.52	2.74	1.33	20.0	21.33	14.67
69	5.0	3.61	5.66	1.81	18.33	6.67	33.33	15.0
70	4.33	3.06	4.83	1.68	8.67	13.0	18.78	5.78
71	6.67	4.16	7.51	2.04	22.22	8.89	6.67	33.33
72	2.67	2.08	3.01	1.66	1.78	19.56	5.33	4.44
73	3.33	1.53	3.75	2.94	13.33	1.11	2.22	8.89
74	8.0	1.0	8.37	11.31	24.0	13.33	29.33	58.67
75	7.0	3.0	7.85	3.25	14.0	11.67	37.33	16.33
76	4.33	2.52	4.92	2.3	11.56	13.0	24.56	14.44
77	6.0	4.58	6.54	1.42	10.0	34.0	30.0	2.0
78	6.67	4.16	7.51	2.04	20.0	15.56	22.22	28.89
79	4.0	3.61	4.38	1.36	9.33	14.67	14.67	12.0
80	4.33	3.51	4.79	1.47	2.89	26.0	2.89	7.22
81	5.67	4.51	6.18	1.4	18.89	26.44	22.67	13.22
82	6.0	4.58	6.54	1.42	24.0	8.0	44.0	28.0
83	6.0	2.65	6.75	3.02	28.0	22.0	18.0	32.0
84	4.0	3.46	4.49	1.53	22.67	12.0	2.67	4.0
85	6.67	1.53	7.18	6.69	20.0	31.11	17.78	33.33
86	5.0	3.61	5.66	1.81	1.67	28.33	15.0	18.33
87	7.33	1.15	7.73	11.1	22.0	29.33	2.44	19.56
88	4.0	3.61	4.38	1.36	6.67	18.67	16.0	6.67
89	5.0	3.61	5.66	1.81	21.67	16.67	13.33	28.33
90	8.0	2.65	8.81	4.73	5.33	34.67	16.0	24.0
91	2.67	0.58	2.86	7.87	7.11	13.33	8.0	11.56
92	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A.13: Start times of tasks in first version of first project at level 0% uncertainly

.

Task	Simulated			Longer First			Shorter First		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
2	8	4	12	8	0	8	8	0	8
3	4	0	4	4	18	22	4	17	21
4	6	0	6	6	0	6	6	0	6
5	3	9	12	3	6	9	3	6	9
6	8	31	39	8	8	16	8	8	16
7	5	4	9	5	21	26	5	21	26
8	9	4	13	9	21	30	9	21	30
9	2	12	14	2	8	10	2	8	10
10	7	6	13	7	6	13	7	6	13
11	9	12	21	9	8	17	9	8	17
12	2	13	15	2	30	32	2	30	32
13	6	4	10	6	21	27	6	21	27
14	3	16	19	3	32	35	3	32	35
15	9	12	21	9	8	17	9	8	17
16	10	13	23	10	16	26	10	16	26
17	6	23	29	6	41	47	6	41	47
18	5	10	15	5	28	33	5	28	33
19	3	13	16	3	30	33	3	30	33
20	7	26	33	7	35	42	7	35	42
21	2	29	31	2	26	28	2	26	28
22	7	29	36	7	47	54	7	47	54
23	2	36	38	2	54	56	2	54	56
24	3	38	41	3	56	59	3	56	59
25	3	33	36	3	42	45	3	42	45
26	7	21	28	7	17	24	7	17	24
27	8	15	23	8	33	41	8	33	41
28	3	33	36	3	49	52	3	49	52
29	7	19	26	7	42	49	7	42	49
30	2	41	43	2	59	61	2	59	61
31	2	36	38	2	52	54	2	52	54
Project Duration	38			54			54		

Table A.14: Start times of tasks in second version of first project at level 0% uncertainly.

Task	Simulated			Longer First			Shorter First		
	Duration	**Start**	**Finish**	**Duration**	**Start**	**Finish**	**Duration**	**Start**	**Finish**
2	8	0	8	8	3	11	8	0	8
3	5	0	5	5	2	7	5	0	5
4	9	0	9	9	4	13	9	0	9
5	3	8	11	3	11	14	3	8	11
6	6	11	17	6	15	21	6	12	18
7	8	17	25	8	20	28	8	19	27
8	8	9	17	8	11	19	8	9	17
9	10	25	35	10	28	38	10	26	36
10	1	5	6	1	8	9	1	5	6
11	5	6	11	5	9	14	5	7	12
12	7	9	16	7	11	18	7	11	18
13	10	6	16	10	9	19	10	7	17
14	3	17	20	3	19	22	3	19	22
15	2	35	37	2	39	41	2	36	38
16	6	11	17	6	15	21	6	12	18
17	4	17	21	4	21	25	4	17	21
18	5	17	22	5	21	26	5	17	22
19	6	22	28	6	25	31	6	22	28
20	6	17	23	6	21	27	6	18	24
21	7	23	30	7	25	32	7	23	30
22	3	20	23	3	23	26	3	22	25
23	8	35	43	8	39	47	8	37	45
24	8	30	38	8	34	42	8	32	40
25	4	25	29	4	28	32	4	27	31
26	2	21	23	2	24	26	2	21	23
27	3	23	26	3	26	29	3	24	27
28	5	38	43	5	41	46	5	40	45
29	3	43	46	3	47	50	3	44	47
30	2	37	39	2	40	42	2	38	40
31	8	43	51	8	46	54	8	45	53
Project Duration	51			54			53		

Table A.15: Start times of tasks in third version of first project at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	7	0	7	7	3	10	7	0	7
3	10	0	10	10	3	13	10	0	10
4	2	0	2	2	5	7	2	0	2
5	6	7	13	6	12	18	6	8	14
6	8	13	21	8	15	23	8	13	21
7	1	10	11	1	15	16	1	11	12
8	8	21	29	8	27	35	8	21	29
9	1	13	14	1	15	16	1	16	17
10	7	13	20	7	18	25	7	17	24
11	4	14	18	4	20	24	4	18	22
12	6	20	26	6	22	28	6	24	30
13	4	26	30	4	29	33	4	26	30
14	9	26	35	9	29	38	9	27	36
15	10	11	21	10	14	24	10	14	24
16	1	30	31	1	32	33	1	32	33
17	7	31	38	7	36	43	7	34	41
18	8	29	37	8	32	40	8	31	39
19	6	21	27	6	26	32	6	21	27
20	2	21	23	2	26	28	2	22	24
21	5	11	16	5	13	18	5	15	20
22	6	27	33	6	33	39	6	27	33
23	2	35	37	2	37	39	2	35	37
24	2	37	39	2	42	44	2	38	40
25	6	37	43	6	42	48	6	41	47
26	9	37	46	9	43	52	9	38	47
27	4	35	39	4	39	43	4	38	42
28	10	37	47	10	42	52	10	38	48
29	3	39	42	3	41	44	3	39	42
30	5	47	52	5	53	58	5	50	55
31	8	46	54	8	51	59	8	50	58
Project Duration	54			59			58		

Table A.16: Start times of tasks in first version of second project at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	8	0	8	8	0	8	8	0	8
3	1	0	1	1	0	1	1	0	1
4	10	0	10	10	0	10	10	0	10
5	6	10	16	6	10	16	6	10	16
6	5	16	21	5	16	21	5	16	21
7	8	8	16	8	8	16	8	8	16
8	9	10	19	9	10	19	9	10	19
9	1	25	26	1	35	36	1	25	26
10	9	10	19	9	13	22	9	10	19
11	8	19	27	8	25	33	8	19	27
12	3	19	22	3	10	13	3	22	25
13	6	26	32	6	39	45	6	27	33
14	2	1	3	2	1	3	2	1	3
15	5	8	13	5	8	13	5	8	13
16	1	16	17	1	16	17	1	16	17
17	3	25	28	3	21	24	3	21	24
18	10	32	42	10	45	55	10	33	43
19	9	16	25	9	16	25	9	16	25
20	1	19	20	1	19	20	1	19	20
21	3	22	25	3	19	22	3	25	28
22	6	16	22	6	16	22	6	16	22
23	3	16	19	3	16	19	3	16	19
24	3	33	36	3	22	25	3	19	22
25	7	13	20	7	13	20	7	13	20
26	6	32	38	6	33	39	6	33	39
27	10	22	32	10	25	35	10	25	35
28	9	42	51	9	55	64	9	43	52
29	8	1	9	8	1	9	8	1	9
30	4	32	36	4	35	39	4	42	46
31	3	23	26	3	25	28	3	24	27
32	3	30	33	3	25	28	3	26	29
33	6	42	48	6	55	61	6	43	49
34	1	3	4	1	3	4	1	3	4
35	9	26	35	9	36	45	9	26	35
36	9	38	47	9	45	54	9	39	48
37	1	32	33	1	35	36	1	35	36
38	2	21	23	2	24	26	2	25	27
39	4	48	52	4	61	65	4	49	53
40	9	32	41	9	20	29	9	26	35
41	10	9	19	10	9	19	10	9	19
42	8	47	55	8	54	62	8	48	56
43	4	28	32	4	28	32	4	28	32
44	3	27	30	3	33	36	3	27	30
45	6	26	32	6	29	35	6	35	41
46	6	19	25	6	19	25	6	19	25
47	7	51	58	7	64	71	7	52	59
48	3	55	58	3	62	65	3	56	59
49	2	38	40	2	39	41	2	39	41

50	10	36	46	10	39	49	10	32	42
51	4	33	37	4	36	40	4	36	40
52	2	58	60	2	71	73	2	59	61
53	1	52	53	1	65	66	1	53	54
54	4	60	64	4	73	77	4	61	65
55	10	64	74	10	77	87	10	65	75
56	8	53	61	8	66	74	8	54	62
57	6	74	80	6	87	93	6	75	81
58	10	46	56	10	49	59	10	46	56
59	3	80	83	3	93	96	3	81	84
60	10	40	50	10	41	51	10	41	51
61	1	61	62	1	74	75	1	62	63
Project Duration	62			75			63		

Table A.17: Start times of tasks in second version of second at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	8	0	8	8	2	10	8	0	8
3	2	0	2	2	4	6	2	2	4
4	3	0	3	3	4	7	3	1	4
5	5	3	8	5	5	10	5	3	8
6	1	3	4	1	6	7	1	5	6
7	5	3	8	5	5	10	5	4	9
8	9	8	17	9	10	19	9	10	19
9	9	8	17	9	11	20	9	10	19
10	5	2	7	5	6	11	5	2	7
11	6	8	14	6	11	17	6	8	14
12	9	14	23	9	16	25	9	15	24
13	1	17	18	1	20	21	1	18	19
14	6	23	29	6	27	33	6	24	30
15	4	8	12	4	11	15	4	9	13
16	10	8	18	10	11	21	10	9	19
17	6	7	13	6	9	15	6	7	13
18	4	2	6	4	5	9	4	2	6
19	8	18	26	8	21	29	8	20	28
20	7	26	33	7	30	37	7	28	35
21	4	17	21	4	19	23	4	17	21
22	3	13	16	3	15	18	3	13	16
23	3	26	29	3	30	33	3	27	30
24	1	29	30	1	32	33	1	29	30
25	9	17	26	9	21	30	9	19	28
26	10	12	22	10	15	25	10	12	22
27	6	29	35	6	33	39	6	29	35
28	3	16	19	3	19	22	3	18	21
29	10	29	39	10	31	41	10	30	40
30	9	29	38	9	33	42	9	30	39
31	4	30	34	4	32	36	4	30	34
32	1	8	9	1	12	13	1	8	9
33	1	9	10	1	11	12	1	11	12
34	10	39	49	10	43	53	10	40	50
35	5	22	27	5	24	29	5	24	29
36	8	21	29	8	24	32	8	21	29
37	6	29	35	6	33	39	6	31	37
38	3	35	38	3	38	41	3	37	40
39	8	38	46	8	42	50	8	39	47
40	2	49	51	2	53	55	2	49	51
41	2	26	28	2	29	31	2	26	28
42	8	30	38	8	34	42	8	30	38
43	1	38	39	1	40	41	1	40	41
44	8	51	59	8	54	62	8	53	61
45	8	21	29	8	23	31	8	22	30
46	4	51	55	4	53	57	4	53	57
47	5	35	40	5	37	42	5	37	42
48	4	40	44	4	43	47	4	40	44
49	4	38	42	4	42	46	4	38	42
50	6	59	65	6	62	68	6	61	67
51	9	49	58	9	51	60	9	51	60

52	8	27	35	8	29	37	8	29	37
53	3	44	47	3	46	49	3	45	48
54	1	58	59	1	61	62	1	60	61
55	6	65	71	6	68	74	6	67	73
56	7	42	49	7	45	52	7	43	50
57	2	55	57	2	59	61	2	55	57
58	4	33	37	4	36	40	4	33	37
59	5	71	76	5	75	80	5	71	76
60	1	49	50	1	51	52	1	49	50
61	6	49	55	6	52	58	6	51	57
Project Duration	55			58			57		

Table A.18: Start times of tasks in third version of second project at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	1	0	1	1	5	6	1	2	3
3	9	0	9	9	3	12	9	2	11
4	3	0	3	3	6	9	3	2	5
5	6	1	7	6	5	11	6	2	8
6	7	1	8	7	5	12	7	4	11
7	4	1	5	4	6	10	4	4	8
8	1	7	8	1	12	13	1	9	10
9	3	8	11	3	10	13	3	12	15
10	4	7	11	4	13	17	4	10	14
11	2	7	9	2	10	12	2	10	12
12	10	11	21	10	13	23	10	12	22
13	2	9	11	2	14	16	2	12	14
14	1	21	22	1	27	28	1	23	24
15	5	9	14	5	12	17	5	13	18
16	4	21	25	4	24	28	4	24	28
17	4	21	25	4	25	29	4	23	27
18	10	8	18	10	11	21	10	10	20
19	9	5	14	9	8	17	9	9	18
20	8	11	19	8	13	21	8	13	21
21	2	14	16	2	17	19	2	14	16
22	10	3	13	10	6	16	10	3	13
23	7	22	29	7	25	32	7	25	32
24	4	22	26	4	25	29	4	22	26
25	1	25	26	1	30	31	1	26	27
26	3	11	14	3	17	20	3	15	18
27	8	18	26	8	20	28	8	22	30
28	4	14	18	4	19	23	4	14	18
29	2	29	31	2	32	34	2	31	33
30	3	22	25	3	28	31	3	25	28
31	6	14	20	6	17	23	6	16	22
32	3	26	29	3	32	35	3	29	32
33	4	31	35	4	33	37	4	34	38
34	10	29	39	10	32	42	10	32	42
35	3	29	32	3	35	38	3	33	36
36	4	26	30	4	29	33	4	27	31
37	10	26	36	10	32	42	10	30	40
38	3	29	32	3	32	35	3	32	35
39	5	31	36	5	34	39	5	34	39
40	6	32	38	6	36	42	6	34	40
41	7	13	20	7	15	22	7	13	20
42	9	36	45	9	42	51	9	40	49
43	3	38	41	3	43	46	3	42	45
44	2	30	32	2	36	38	2	32	34
45	2	36	38	2	39	41	2	38	40
46	4	38	42	4	44	48	4	42	46
47	8	32	40	8	34	42	8	32	40
48	4	45	49	4	49	53	4	49	53
49	4	38	42	4	41	45	4	40	44

50	2	16	18	2	18	20	2	18	20
51	2	39	41	2	41	43	2	43	45
52	9	42	51	9	47	56	9	44	53
53	8	51	59	8	56	64	8	51	59
54	2	45	47	2	48	50	2	48	50
55	10	42	52	10	48	58	10	42	52
56	4	45	49	4	50	54	4	45	49
57	10	52	62	10	58	68	10	56	66
58	10	41	51	10	47	57	10	45	55
59	8	62	70	8	68	76	8	63	71
60	4	47	51	4	52	56	4	51	55
61	8	49	57	8	52	60	8	50	58
Project Duration	57			60			58		

Table A.19: Start times of tasks in first version of third project at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	8	0	8	8	0	8	8	0	8
3	10	0	10	10	0	10	10	0	10
4	1	0	1	1	0	1	1	0	1
5	2	1	3	2	1	3	2	1	3
6	10	1	11	10	1	11	10	10	20
7	8	13	21	8	11	19	8	11	19
8	6	11	17	6	18	24	6	11	17
9	9	11	20	9	11	20	9	21	30
10	1	3	4	1	3	4	1	3	4
11	8	8	16	8	8	16	8	8	16
12	2	19	21	2	19	21	2	16	18
13	7	4	11	7	4	11	7	4	11
14	2	30	32	2	33	35	2	19	21
15	7	17	24	7	11	18	7	17	24
16	4	21	25	4	21	25	4	18	22
17	10	20	30	10	20	30	10	30	40
18	9	21	30	9	21	30	9	19	28
19	9	10	19	9	10	19	9	1	10
20	3	24	27	3	20	23	3	24	27
21	7	32	39	7	30	37	7	31	38
22	2	11	13	2	19	21	2	21	23
23	9	24	33	9	24	33	9	24	33
24	6	8	14	6	8	14	6	8	14
25	3	20	23	3	20	23	3	30	33
26	3	11	14	3	11	14	3	20	23
27	6	26	32	6	33	39	6	23	29
28	5	29	34	5	39	44	5	29	34
29	6	20	26	6	19	25	6	31	37
30	8	3	11	8	3	11	8	3	11
31	5	14	19	5	14	19	5	14	19
32	4	39	43	4	37	41	4	39	43
33	7	27	34	7	23	30	7	27	34
34	7	33	40	7	33	40	7	33	40
35	2	33	35	2	33	35	2	35	37
36	8	24	32	8	18	26	8	34	42
37	4	16	20	4	16	20	4	41	45
38	9	34	43	9	26	35	9	22	31
39	5	14	19	5	14	19	5	34	39
40	7	32	39	7	26	33	7	42	49
41	6	39	45	6	44	50	6	36	42
42	8	41	49	8	25	33	8	37	45
43	8	33	41	8	33	41	8	42	50
44	1	26	27	1	25	26	1	39	40
45	1	19	20	1	58	59	1	45	46
46	8	21	29	8	21	29	8	46	54
47	6	42	48	6	35	41	6	34	40
48	9	32	41	9	29	38	9	43	52
49	1	38	39	1	39	40	1	29	30
50	8	49	57	8	38	46	8	52	60
51	8	44	52	8	44	52	8	46	54

52	10	35	45	10	35	45	10	37	47
53	3	34	37	3	38	41	3	40	43
54	6	48	54	6	41	47	6	40	46
55	3	39	42	3	37	40	3	45	48
56	10	34	44	10	46	56	10	21	31
57	5	49	54	5	41	46	5	54	59
58	2	52	54	2	56	58	2	54	56
59	3	54	57	3	58	61	3	56	59
60	3	48	51	3	56	59	3	38	41
61	3	48	51	3	52	55	3	47	50
62	5	43	48	5	40	45	5	48	53
63	9	42	51	9	44	53	9	50	59
64	5	48	53	5	45	50	5	53	58
65	4	30	34	4	30	34	4	40	44
66	8	30	38	8	30	38	8	28	36
67	9	14	23	9	14	23	9	14	23
68	4	59	63	4	59	63	4	60	64
69	2	45	47	2	56	58	2	33	35
70	7	51	58	7	53	60	7	59	66
71	2	43	45	2	41	43	2	46	48
72	1	58	59	1	63	64	1	66	67
73	5	47	52	5	58	63	5	47	52
74	9	51	60	9	59	68	9	50	59
75	4	54	58	4	52	56	4	58	62
76	2	54	56	2	58	60	2	56	58
77	7	51	58	7	45	52	7	53	60
78	8	63	71	8	63	71	8	64	72
79	8	60	68	8	68	76	8	60	68
80	1	60	61	1	56	57	1	68	69
81	6	59	65	6	64	70	6	67	73
82	10	65	75	10	57	67	10	69	79
83	9	64	73	9	68	77	9	59	68
84	2	71	73	2	71	73	2	72	74
85	7	57	64	7	61	68	7	68	75
86	9	58	67	9	52	61	9	60	69
87	8	68	76	8	76	84	8	77	85
88	1	76	77	1	84	85	1	72	73
89	2	77	79	2	85	87	2	85	87
90	9	59	68	9	64	73	9	67	76
91	3	76	79	3	73	76	3	74	77
Project Duration	79			76			77		

Table A.20: Start times of tasks in second version of third project at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	9	0	9	9	2	11	9	0	9
3	2	0	2	2	2	4	2	2	4
4	6	0	6	6	3	9	6	0	6
5	8	6	14	8	8	16	8	7	15
6	1	14	15	1	18	19	1	16	17
7	3	14	17	3	18	21	3	15	18
8	1	14	15	1	17	18	1	15	16
9	6	15	21	6	17	23	6	17	23
10	8	21	29	8	23	31	8	21	29
11	6	9	15	6	13	19	6	9	15
12	4	29	33	4	32	36	4	30	34
13	7	15	22	7	18	25	7	15	22
14	8	15	23	8	18	26	8	17	25
15	4	15	19	4	18	22	4	16	20
16	6	33	39	6	35	41	6	35	41
17	8	29	37	8	33	41	8	31	39
18	10	9	19	10	13	23	10	9	19
19	6	15	21	6	19	25	6	16	22
20	4	15	19	4	19	23	4	15	19
21	7	17	24	7	19	26	7	18	25
22	1	21	22	1	25	26	1	21	22
23	2	19	21	2	21	23	2	20	22
24	8	15	23	8	17	25	8	17	25
25	3	19	22	3	22	25	3	21	24
26	1	22	23	1	24	25	1	22	23
27	1	24	25	1	27	28	1	24	25
28	8	22	30	8	25	33	8	22	30
29	9	19	28	9	23	32	9	19	28
30	10	2	12	10	5	15	10	4	14
31	3	28	31	3	32	35	3	29	32
32	4	29	33	4	33	37	4	30	34
33	7	37	44	7	39	46	7	37	44
34	1	31	32	1	34	35	1	31	32
35	9	19	28	9	22	31	9	20	29
36	8	6	14	8	8	16	8	6	14
37	1	37	38	1	39	40	1	37	38
38	4	23	27	4	25	29	4	25	29
39	2	21	23	2	24	26	2	22	24
40	9	23	32	9	25	34	9	24	33
41	3	23	26	3	26	29	3	24	27
42	2	39	41	2	42	44	2	41	43
43	7	38	45	7	42	49	7	38	45
44	1	9	10	1	13	14	1	9	10
45	3	22	25	3	25	28	3	24	27
46	2	23	25	2	25	27	2	23	25
47	8	14	22	8	18	26	8	15	23
48	5	22	27	5	24	29	5	23	28
49	6	27	33	6	31	37	6	27	33

50	3	33	36	3	37	40	3	34	37
51	6	23	29	6	26	32	6	25	31
52	1	23	24	1	26	27	1	25	26
53	3	33	36	3	36	39	3	34	37
54	9	33	42	9	35	44	9	33	42
55	1	41	42	1	45	46	1	42	43
56	8	45	53	8	48	56	8	46	54
57	2	21	23	2	23	25	2	22	24
58	9	33	42	9	37	46	9	35	44
59	7	28	35	7	30	37	7	29	36
60	7	41	48	7	44	51	7	42	49
61	10	6	16	10	8	18	10	6	16
62	3	29	32	3	31	34	3	30	33
63	2	25	27	2	28	30	2	26	28
64	5	24	29	5	27	32	5	26	31
65	8	16	24	8	18	26	8	18	26
66	5	45	50	5	48	53	5	47	52
67	8	42	50	8	44	52	8	42	50
68	2	48	50	2	50	52	2	50	52
69	4	33	37	4	36	40	4	35	39
70	1	50	51	1	54	55	1	51	52
71	10	50	60	10	52	62	10	52	62
72	5	60	65	5	62	67	5	61	66
73	3	36	39	3	38	41	3	37	40
74	7	29	36	7	33	40	7	29	36
75	7	50	57	7	53	60	7	50	57
76	7	38	45	7	40	47	7	40	47
77	10	60	70	10	63	73	10	61	71
78	10	70	80	10	73	83	10	71	81
79	3	50	53	3	52	55	3	52	55
80	8	57	65	8	59	67	8	58	66
81	1	45	46	1	47	48	1	45	46
82	7	50	57	7	54	61	7	52	59
83	5	39	44	5	41	46	5	39	44
84	8	65	73	8	68	76	8	65	73
85	5	57	62	5	60	65	5	57	62
86	2	73	75	2	77	79	2	73	75
87	6	42	48	6	44	50	6	44	50
88	8	62	70	8	66	74	8	62	70
89	9	65	74	9	68	77	9	67	76
90	5	80	85	5	84	89	5	81	86
91	3	75	78	3	79	82	3	76	79
Project Duration	78			82			79		

Table A.21: Start times of tasks in third version of third project. at level 0% uncertainly

Task	Simulated			Longer First			Shorter First		
	Duration	Start	Finish	Duration	Start	Finish	Duration	Start	Finish
2	2	0	2	2	3	5	2	2	4
3	7	0	7	7	4	11	7	0	7
4	7	0	7	7	3	10	7	2	9
5	10	7	17	10	11	21	10	7	17
6	8	7	15	8	11	19	8	11	19
7	6	15	21	6	21	27	6	15	21
8	9	21	30	9	24	33	9	23	32
9	3	7	10	3	10	13	3	8	11
10	7	21	28	7	23	30	7	23	30
11	3	7	10	3	10	13	3	10	13
12	3	15	18	3	21	24	3	19	22
13	2	30	32	2	32	34	2	30	32
14	4	28	32	4	30	34	4	32	36
15	10	30	40	10	33	43	10	33	43
16	10	21	31	10	27	37	10	25	35
17	10	15	25	10	18	28	10	19	29
18	8	32	40	8	38	46	8	36	44
19	1	18	19	1	22	23	1	21	22
20	6	40	46	6	42	48	6	44	50
21	4	46	50	4	50	54	4	50	54
22	9	30	39	9	35	44	9	33	42
23	3	40	43	3	42	45	3	40	43
24	10	28	38	10	31	41	10	32	42
25	4	17	21	4	21	25	4	21	25
26	1	39	40	1	43	44	1	42	43
27	1	40	41	1	44	45	1	42	43
28	8	18	26	8	24	32	8	21	29
29	6	31	37	6	33	39	6	32	38
30	4	32	36	4	38	42	4	35	39
31	5	10	15	5	15	20	5	11	16
32	9	43	52	9	47	56	9	44	53
33	6	28	34	6	34	40	6	30	36
34	5	7	12	5	12	17	5	8	13
35	8	10	18	8	15	23	8	12	20
36	9	19	28	9	24	33	9	23	32
37	10	21	31	10	24	34	10	22	32
38	4	43	47	4	48	52	4	45	49
39	3	46	49	3	52	55	3	49	52
40	5	38	43	5	41	46	5	38	43
41	8	36	44	8	38	46	8	40	48
42	4	43	47	4	48	52	4	45	49
43	1	49	50	1	51	52	1	52	53
44	9	52	61	9	58	67	9	56	65
45	9	50	59	9	55	64	9	53	62
46	6	40	46	6	44	50	6	41	47
47	4	32	36	4	36	40	4	36	40
48	9	44	53	9	47	56	9	46	55
49	2	26	28	2	30	32	2	27	29

50	3	52	55	3	55	58	3	56	59
51	4	36	40	4	42	46	4	36	40
52	10	43	53	10	48	58	10	46	56
53	6	53	59	6	59	65	6	55	61
54	3	47	50	3	50	53	3	47	50
55	7	59	66	7	61	68	7	63	70
56	6	41	47	6	47	53	6	43	49
57	3	46	49	3	48	51	3	47	50
58	2	61	63	2	65	67	2	62	64
59	9	46	55	9	50	59	9	50	59
60	9	40	49	9	46	55	9	42	51
61	9	49	58	9	55	64	9	51	60
62	3	44	47	3	47	50	3	46	49
63	9	53	62	9	55	64	9	57	66
64	10	52	62	10	54	64	10	56	66
65	5	49	54	5	51	56	5	53	58
66	2	40	42	2	42	44	2	41	43
67	7	42	49	7	45	52	7	43	50
68	6	58	64	6	64	70	6	61	67
69	9	53	62	9	58	67	9	54	63
70	5	58	63	5	61	66	5	60	65
71	8	66	74	8	71	79	8	70	78
72	2	42	44	2	47	49	2	43	45
73	2	54	56	2	59	61	2	56	58
74	8	74	82	8	77	85	8	78	86
75	10	50	60	10	55	65	10	53	63
76	4	60	64	4	62	66	4	60	64
77	1	47	48	1	53	54	1	51	52
78	2	63	65	2	69	71	2	65	67
79	1	55	56	1	61	62	1	59	60
80	4	82	86	4	84	88	4	84	88
81	10	59	69	10	63	73	10	62	72
82	1	65	66	1	68	69	1	67	68
83	4	69	73	4	73	77	4	69	73
84	2	66	68	2	70	72	2	70	72
85	8	64	72	8	68	76	8	68	76
86	4	49	53	4	55	59	4	49	53
87	8	64	72	8	68	76	8	67	75
88	3	86	89	3	91	94	3	86	89
89	4	89	93	4	94	98	4	91	95
90	10	68	78	10	72	82	10	70	80
91	2	72	74	2	74	76	2	76	78
Project Duration	74			76			78		

Appendix B

```
clear all;
clc;

global config;

config.uncertainty_level = 0.05;
config.num_samples = 100;
config.num_std = 2;
config.proposed = 0;

nvars = 150;
PopulationSize_Data = 20;
EliteCount_Data = 4;
CrossoverFraction_Data = 0.7;
Generations_Data = 10;

project = createJ30smProject();
project = calculatePredecessors(project);

num_expr = 0;
config.num_std = 2;
for i = 1:5

    num_expr = num_expr + 1
    config.uncertainty_level = i*0.05;
    x_axis(i) = config.uncertainty_level;

    config.proposed = 0;
    [x,fval,exitflag,output,population,score] =
gaJ30smProject(nvars,PopulationSize_Data,EliteCount_Data,Cross
overFraction_Data,Generations_Data);
    life_span_not_proposed(i) =
fitnessJ30smProjectValidation(x);
    tasks_order = decodeChromosomeToTasksOrder(x,project);
    average_start_time_not_proposed(i) =
```

```
getAverageStartTime(project,tasks_order);
    average_finish_time_not_proposed(i) =
getAverageFinishTime(project,tasks_order);
```

```
    num_expr = num_expr + 1
    config.proposed = 1;
    [x,fval,exitflag,output,population,score] =
gaJ30smProject(nvars,PopulationSize_Data,EliteCount_Data,Cross
overFraction_Data,Generations_Data);
    life_span(i) = fitnessJ30smProjectValidation(x);
    tasks_order = decodeChromosomeToTasksOrder(x,project);
    average_start_time(i) =
getAverageStartTime(project,tasks_order);
    average_finish_time(i) =
getAverageFinishTime(project,tasks_order);
```

```
end
```

```
trace_95.x_axis = x_axis;

trace_95.life_span_not_proposed = life_span_not_proposed;
trace_95.average_start_time_not_proposed =
average_start_time_not_proposed;
trace_95.average_finish_time_not_proposed =
average_finish_time_not_proposed;

trace_95.life_span = life_span;
trace_95.average_start_time = average_start_time;
trace_95.average_finish_time = average_finish_time;
```

```
save('trace_95.mat','trace_95');
```

```
config.num_std = 3;
for i = 1:5
```

```
    num_expr = num_expr + 1
    config.uncertainty_level = i*0.05;
    x_axis(i) = config.uncertainty_level;
```

```
    config.proposed = 0;
    [x,fval,exitflag,output,population,score] =
gaJ30smProject(nvars,PopulationSize_Data,EliteCount_Data,Cross
overFraction_Data,Generations_Data);
    life_span_not_proposed(i) =
fitnessJ30smProjectValidation(x);
    tasks_order = decodeChromosomeToTasksOrder(x,project);
    average_start_time_not_proposed(i) =
getAverageStartTime(project,tasks_order);
    average_finish_time_not_proposed(i) =
getAverageFinishTime(project,tasks_order);
```

```
    num_expr = num_expr + 1
    config.proposed = 1;
    [x,fval,exitflag,output,population,score] =
gaJ30smProject(nvars,PopulationSize_Data,EliteCount_Data,Cross
overFraction_Data,Generations_Data);
    life_span(i) = fitnessJ30smProjectValidation(x);
    tasks_order = decodeChromosomeToTasksOrder(x,project);
    average_start_time(i) =
getAverageStartTime(project,tasks_order);
    average_finish_time(i) =
getAverageFinishTime(project,tasks_order);
```

```
end
```

```
trace_99.x_axis = x_axis;

trace_99.life_span_not_proposed = life_span_not_proposed;
trace_99.average_start_time_not_proposed =
average_start_time_not_proposed;
trace_99.average_finish_time_not_proposed =
average_finish_time_not_proposed;

trace_99.life_span = life_span;
trace_99.average_start_time = average_start_time;
trace_99.average_finish_time = average_finish_time;
```

```
save('trace_99.mat','trace_99');
```

```
function life_span = getLifeSpan(project,tasks_order)
clock = 0;
completed_tasks(1) = 1;
scheduled_tasks = [];
remaining_tasks = tasks_order(2:end);
resource_availability = project.resource_availability;
start_events = [];
finish_events = [];

project_complated = 0;

while project_complated == 0
    num_of_new_scheduled_task = 0;
    i = 1;
    while i <= size(remaining_tasks,2)
        cannot_be_started = 0;
        %disp(i);
        for j =
1:project.tasks(remaining_tasks(i)).num_predecessors
            task_exists = 0;
            for k = 1:size(completed_tasks,2)
                if
project.tasks(remaining_tasks(i)).predecessors(j) ==
completed_tasks(k)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            if min(resource_availability -
project.tasks(remaining_tasks(i)).usage_of_resource) >= 0
                scheduled_tasks(size(scheduled_tasks,2)+1) =
remaining_tasks(i);
```

```
                start_events(size(start_events,1)+1,1) =
remaining_tasks(i);
                start_events(size(start_events,1),2) = clock;
                finish_events(size(finish_events,1)+1,1) =
remaining_tasks(i);
                finish_events(size(finish_events,1),2) = clock
+ project.tasks(remaining_tasks(i)).duration;
                resource_availability = resource_availability
- project.tasks(remaining_tasks(i)).usage_of_resource;
                remaining_tasks(i) = [];
                num_of_new_scheduled_task =
num_of_new_scheduled_task + 1;
            end
        end
        i = i + 1;
    end
    if num_of_new_scheduled_task == 0
        finish_events_temp = [];
        for i = 1:size(finish_events,1)
            task_finished = 0;
            for j = 1:size(completed_tasks,2)
                if finish_events(i,1) == completed_tasks(j)
                    task_finished = 1;
                end
            end
            if task_finished == 0

finish_events_temp(size(finish_events_temp,1)+1,1) =
finish_events(i,1);

finish_events_temp(size(finish_events_temp,1),2) =
finish_events(i,2);
            end
        end
        time_diff = inf;
        temp_index = 0;
        for i = 1:size(finish_events_temp,1)
```

```
            if time_diff > finish_events_temp(i,2) - clock
                time_diff = finish_events_temp(i,2) - clock;
                temp_index = i;
            end
        end
        resource_availability = resource_availability +
project.tasks(finish_events_temp(temp_index,1)).usage_of_resou
rce;
        clock = finish_events_temp(temp_index,2);
        completed_tasks(size(completed_tasks,2)+1) =
finish_events_temp(temp_index,1);
    end
    if remaining_tasks(1) == project.num_tasks
        project_complated = 1;
    end
end
life_span = max(finish_events(:,2));
end
```

```
function average_start_time =
getAverageStartTime(project,tasks_order)
clock = 0;
completed_tasks(1) = 1;
scheduled_tasks = [];
remaining_tasks = tasks_order(2:end);
resource_availability = project.resource_availability;
start_events = [];
finish_events = [];

project_complated = 0;

while project_complated == 0
    num_of_new_scheduled_task = 0;
    i = 1;
    while i <= size(remaining_tasks,2)
        cannot_be_started = 0;
        %disp(i);
        for j =
1:project.tasks(remaining_tasks(i)).num_predecessors
            task_exists = 0;
            for k = 1:size(completed_tasks,2)
                if
project.tasks(remaining_tasks(i)).predecessors(j) ==
completed_tasks(k)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            if min(resource_availability -
project.tasks(remaining_tasks(i)).usage_of_resource) >= 0
                scheduled_tasks(size(scheduled_tasks,2)+1) =
```

```
remaining_tasks(i);
                start_events(size(start_events,1)+1,1) =
remaining_tasks(i);
                start_events(size(start_events,1),2) = clock;
                finish_events(size(finish_events,1)+1,1) =
remaining_tasks(i);
                finish_events(size(finish_events,1),2) = clock
+ project.tasks(remaining_tasks(i)).duration;
                resource_availability = resource_availability
- project.tasks(remaining_tasks(i)).usage_of_resource;
                remaining_tasks(i) = [];
                num_of_new_scheduled_task =
num_of_new_scheduled_task + 1;
            end
        end
        i = i + 1;
    end
    if num_of_new_scheduled_task == 0
        finish_events_temp = [];
        for i = 1:size(finish_events,1)
            task_finished = 0;
            for j = 1:size(completed_tasks,2)
                if finish_events(i,1) == completed_tasks(j)
                    task_finished = 1;
                end
            end
            if task_finished == 0

finish_events_temp(size(finish_events_temp,1)+1,1) =
finish_events(i,1);

finish_events_temp(size(finish_events_temp,1),2) =
finish_events(i,2);
            end
        end
        time_diff = inf;
        temp_index = 0;
```

```
        for i = 1:size(finish_events_temp,1)
            if time_diff > finish_events_temp(i,2) - clock
                time_diff = finish_events_temp(i,2) - clock;
                temp_index = i;
            end
        end
        resource_availability = resource_availability +
project.tasks(finish_events_temp(temp_index,1)).usage_of_resou
rce;
        clock = finish_events_temp(temp_index,2);
        completed_tasks(size(completed_tasks,2)+1) =
finish_events_temp(temp_index,1);
    end
    if remaining_tasks(1) == project.num_tasks
        project_complated = 1;
    end
end
average_start_time = mean(start_events(:,2));
end
```

```
function average_finish_time =
getAverageFinishTime(project,tasks_order)
clock = 0;
completed_tasks(1) = 1;
scheduled_tasks = [];
remaining_tasks = tasks_order(2:end);
resource_availability = project.resource_availability;
start_events = [];
finish_events = [];

project_complated = 0;

while project_complated == 0
    num_of_new_scheduled_task = 0;
    i = 1;
    while i <= size(remaining_tasks,2)
        cannot_be_started = 0;
        %disp(i);
        for j =
1:project.tasks(remaining_tasks(i)).num_predecessors
            task_exists = 0;
            for k = 1:size(completed_tasks,2)
                if
project.tasks(remaining_tasks(i)).predecessors(j) ==
completed_tasks(k)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            if min(resource_availability -
project.tasks(remaining_tasks(i)).usage_of_resource) >= 0
                scheduled_tasks(size(scheduled_tasks,2)+1) =
```

```
remaining_tasks(i);
                start_events(size(start_events,1)+1,1) =
remaining_tasks(i);
                start_events(size(start_events,1),2) = clock;
                finish_events(size(finish_events,1)+1,1) =
remaining_tasks(i);
                finish_events(size(finish_events,1),2) = clock
+ project.tasks(remaining_tasks(i)).duration;
                resource_availability = resource_availability
- project.tasks(remaining_tasks(i)).usage_of_resource;
                remaining_tasks(i) = [];
                num_of_new_scheduled_task =
num_of_new_scheduled_task + 1;
            end
        end
        i = i + 1;
    end
    if num_of_new_scheduled_task == 0
        finish_events_temp = [];
        for i = 1:size(finish_events,1)
            task_finished = 0;
            for j = 1:size(completed_tasks,2)
                if finish_events(i,1) == completed_tasks(j)
                    task_finished = 1;
                end
            end
            if task_finished == 0

finish_events_temp(size(finish_events_temp,1)+1,1) =
finish_events(i,1);

finish_events_temp(size(finish_events_temp,1),2) =
finish_events(i,2);
            end
        end
        time_diff = inf;
        temp_index = 0;
```

```
        for i = 1:size(finish_events_temp,1)
            if time_diff > finish_events_temp(i,2) - clock
                time_diff = finish_events_temp(i,2) - clock;
                temp_index = i;
            end
        end
        resource_availability = resource_availability +
project.tasks(finish_events_temp(temp_index,1)).usage_of_resou
rce;
        clock = finish_events_temp(temp_index,2);
        completed_tasks(size(completed_tasks,2)+1) =
finish_events_temp(temp_index,1);
    end
    if remaining_tasks(1) == project.num_tasks
        project_complated = 1;
    end
end
average_finish_time = mean(finish_events(:,2));
end
```

```
function chromosome = generateRandomChromosome(project)
num_genes = project.num_tasks - 2;
gene_size = size(dec2bin(num_genes),2);

for i = 1:(num_genes*gene_size)
    if unifrnd(0,1) > 0.5
        chromosome(i) = 1;
    else
        chromosome(i) = 0;
    end
end
end
```

```
function [x,fval,exitflag,output,population,score] =
gaJ30smProject_0_01_uncertainty(nvars,PopulationSize_Data,Elit
eCount_Data,CrossoverFraction_Data,Generations_Data)

options = gaoptimset;

options = gaoptimset(options,'PopulationType', 'bitstring');
options = gaoptimset(options,'PopulationSize',
PopulationSize_Data);
options = gaoptimset(options,'EliteCount', EliteCount_Data);
options = gaoptimset(options,'CrossoverFraction',
CrossoverFraction_Data);
options = gaoptimset(options,'Generations', Generations_Data);
options = gaoptimset(options,'FitnessScalingFcn',
@fitscalingprop);
options = gaoptimset(options,'SelectionFcn',
@selectionroulette);
options = gaoptimset(options,'MutationFcn', {
@mutationuniform 0.05 });
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'PlotFcns', {  @gaplotbestf
@gaplotbestindiv @gaplotdistance @gaplotexpectation
@gaplotgenealogy @gaplotrange @gaplotscorediversity
@gaplotscores @gaplotselection @gaplotstopping
@gaplotmaxconstr });
[x,fval,exitflag,output,population,score] = ...
ga(@fitnessJ30smProject_0_01_uncertainty,nvars,[],[],[],[],[],
[],[],[],options);
```

```
function tasks_order =
decodeChromosomeToTasksOrder(chromosome,project)
num_genes = project.num_tasks - 2;
gene_size = size(dec2bin(num_genes),2);
tasks_order(1) = 1;
tasks_pool = project.tasks(1).successors;
for i = 1:project.num_tasks - 2
    tasks_can_be_started = [];
    tasks_can_be_started_index = [];
    tasks_counter = 0;
    for j = 1:size(tasks_pool,2)
        cannot_be_started = 0;
        for k =
1:project.tasks(tasks_pool(j)).num_predecessors
            task_exists = 0;
            for l = 1:size(tasks_order,2)
                if
project.tasks(tasks_pool(j)).predecessors(k) == tasks_order(l)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            tasks_counter = tasks_counter + 1;
            tasks_can_be_started(tasks_counter) =
tasks_pool(j);
            tasks_can_be_started_index(tasks_counter) = j;
        end
    end
    gene_value = bin2dec(num2str(chromosome( (i-1)*gene_size +
1 : i*gene_size )));
    task_index = mod(gene_value, size(tasks_can_be_started,2))
+ 1;
```

```
    tasks_order(i+1) = tasks_can_be_started(task_index);
    tasks_pool(tasks_can_be_started_index(task_index)) = [];
    tasks_pool =
union(tasks_pool,project.tasks(tasks_order(i+1)).successors);
    for j = 1:project.tasks(tasks_order(i+1)).num_successors
        cannot_be_started = 0;
        for k =
1:project.tasks(project.tasks(tasks_order(i+1)).successors(j))
.num_predecessors
            task_exists = 0;
            for l = 1:size(tasks_order,2)
                if
project.tasks(project.tasks(tasks_order(i+1)).successors(j)).p
redecessors(k) == tasks_order(l)
                    task_exists = 1;
                end
            end
            if task_exists == 0
                cannot_be_started = 1;
            end
        end
        if cannot_be_started == 0
            task_already_exists = 0;
            for k = 1:size(tasks_pool,2)
                if
project.tasks(tasks_order(i+1)).successors(j) == tasks_pool(k)
                    task_already_exists = 1;
                end
            end
            if task_already_exists == 0
                tasks_pool(size(tasks_pool,2)+1) =
project.tasks(tasks_order(i+1)).successors(j);
            end
        end
    end
end
```

```
tasks_order(project.num_tasks) = project.num_tasks;
end
```

```
function project = calculatePredecessors(project)
for task_index = 1:project.num_tasks
    if task_index - 1 == 0
        project.tasks(task_index).num_predecessors = 0;
    else
        counter_predecessors = 0;
        for i = 1:task_index
            for j = 1:project.tasks(i).num_successors
                if project.tasks(i).successors(j) ==
task_index
                    counter_predecessors =
counter_predecessors + 1;

project.tasks(task_index).predecessors(counter_predecessors) =
i;
                end
            end
        end
        project.tasks(task_index).num_predecessors =
counter_predecessors;
    end
end
end
```