OPTIMISING NET PRESENT VALUE USING PRIORITY RULE-BASED SCHEDULING

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2014

Vacharee Tantisuvanichkul

School of Mechanical, Aerospace and Civil Engineering

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

B-PSGS	Backward Parallel Schedule Generation Scheme
B-SSGS	Backward Serial Schedule Generation Scheme
BPV	Net Present value of benefits
CCF	Cumulative Cash Flow
CCPSP	Capital-Constrained Project Scheduling Problem
CCS	Critical Chain Scheduling
CF	Cash flow
CPA	Critical Path Analysis
CPM	Critical Path Method
CPV	Present value of costs
DUR	Duration
EA	Evolutionary algorithms
EFT	Earliest finish time
EST	Earliest start time
F/S	Finish-to-start
F/F	Finish-to-finish
F-PSGS	Forward Parallel Schedule Generation Scheme
F-SSGS	Forward Serial Schedule Generation Scheme
FF	Free float
GA	Genetic Algorithm
GRG	A Generalised Reduced Gradient method
GNS	Great number of successors
IRR	Internal rate of return

LFT	Latest finish time
LP	Linear programming
LST	Latest start time
m-CCF	Modified-cumulative cash flow
MCS	Monte Carlo Simulation
MINSLK	Minimum slack
NLP	Nonlinear programming
NPV	Net present value
PERT	Program Evaluation and Review Technique
PI	Profitability index
PP	Payback Period
PSGS	Parallel Schedule Generation Scheme
PSP	Project scheduling problem
PV	Present Value
RCPSP	Resource constrained project scheduling problem
ROA	Return on assets
ROI	Return on investment
SA	Simulated Annealing
SGS	Schedule Generation Scheme
SSGS	Serial Schedule Generation Scheme
S/F	Start-to-finish
S/S	Start-to-start
TF	Total float
TS	Tabu Search
WACC	Weighted average cost of capital

List of Variables

A_g	Active Set of activity at stage g
A_t	Active Set of activity at period t
α	The discount rate (continuously compounding)
β	The discount factor
C_g	Complete Set of activity at stage g
CF_j	Cash Flow for activity j
CF_t	Cash Flow at period t
d_j	Duration of activity j
D_g	Decision Set of activity at stage g
EFT_j	Earliest finish time of activity j
EST_j	Earliest start time of activity j
FT	Finish Time
I	Capital available
K_g	The remaining of the capital resource at stage g
K_t	The remaining of the capital resource at period t
LFT_j	Latest finish time of activity j
LST_j	Latest start time of activity j
ϕ	An empty set
r	The effective discount rate
m	The number of times compounded
P_j	The set of immediate predecessors of activity j
S_j	Set of successors of activity j

t	Periods
T	The completion time for project
t_g	The schedule time at stage g
t_j	Time at which activity j is scheduled to occur
$v(j)$	A priority value of activity j

Abstract

The University of Manchester

Vacharee Tantisuvanichkul

Doctor of Philosophy (PhD)

Optimising Net Present Value using Priority rule-based scheduling

This research is focused on project scheduling with the aim to capture the monetary objectives of the project in the form of the maximisation of Net Present Value (NPV). In addition, this research is also highlighted key project management practices and scheduling methods. Project scheduling is very attractive for researchers and it has recently been drawn considerable attention because of the high cost of capital and the significant effect of the time value of money. This is the principal motivating factor behind this study.

Project-scheduling problem is solved by priority rule-based heuristic methods in this study. The idea behind heuristic algorithms is to rank the activities by some rules. This research proposes a new rule called m-CCF with improved performance from the existing one. The m-CCF is also embedded in serial and parallel schedule generation schemes and is extended by implementing in a forward and backward strategy. The experiments are conducted to evaluate the performance of the proposed technique measuring the NPV generated for a particular project. This research also presents a framework summarising the previous research on project scheduling techniques. It is found that the m-CCF results in higher NPVs than any other heuristics. A series of different projects are examined to validate the potential of the m-CCF technique. The main findings of the research discover that the m-CCF is worthwhile to be employed in priority rule-based scheduling technique. Furthermore, the main findings suggest that it is beneficial to utilise forward-backward solution for scheduling improvement and selecting the schedule with the largest NPV among those available.

In conclusion, this research contributes to existing knowledge by developing the combination of m-CCF priority rule methods and backward–forward scheduling. This can be considered as a good direction to develop further heuristics that can be exploited as a powerful tool in project planning and control systems.

Declaration

No portion of the work referred to in thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Dedication

This research is dedicated to my family for giving all the love, support and encouragement throughout the duration of my studies.

Acknowledgement

I would like to express my sincere appreciations to the following;

First, I would like to thank my supervisor; Dr. Moray Kidd, for his advice, dedication and carefulness throughout the year.

Second, I would like to thank the members of the school of Mechanical, Aerospace and Civil engineering and Manchester Business School for providing an excellent environment to study and enhance my research.

Thanks go to all staffs of the PTT Public Company Limited for their helps.

Then, I would like to thank all my friends who always care and give good advice during the hard time here.

Finally, I am deeply indebted to my family; parents, aunts and my sister, for their love and support. I could not be here without their encouragement.

September 2013

Vacharee Tantisuvanichkul

Preface

The author is a graduate with an MSc in Engineering Project Management from the University of Manchester and a BEng in Chemical Engineering from Kasetsart University, Bangkok, Thailand.

In September 2009, the author commenced the Doctorate programme, for which this thesis was submitted for examination in September 2013. Approximately two months of this time has been spent working in the PTT Company limited for gaining experience and collecting data for the research. During the four years, the author has produced three peer reviewed conference papers and presented at two international conferences in the UK and Singapore.

The author is in the process of being written a journal paper that will be submitted to the International Journal of Project Management.

Chapter 1: Introduction

1.1 Background

Project management is concerned with the overall planning and co-ordination of a project from conception to completion aimed at meeting the stated requirements and ensuring completion on time, within cost and to required quality standards [1]. The subset of project management that this research will focus on is project scheduling. Project scheduling is concerned with the techniques that can be employed to manage the activities that need to be undertaken during the development of a project [2]. It is primarily concerned with assigning a sequence to activities to be conducted within the project. Project scheduling is very attractive for researchers and has recently attracted considerable attention because of the high cost of capital and the significant effect of the time value of money [3]. Project managers must schedule large projects subject to conflicting objectives and limited resources. Project objectives may include minimising project makespan, efficient utilisation of resources, and effective management of cash outlays and receipts [4, 5]. The constraints and parameters of the project scheduling problem (PSP) include activity durations, precedence relationships, and limits on the availability of capital, labour, materials, facilities, and equipment [1]. Project planners frequently use network scheduling procedures such as the programme evaluation and review technique (PERT) and the critical path method (CPM) to find the duration of the longest path in the network (the critical path) and a feasible schedule [6]. Project managers must also consider the impact of cash flows on the project plan, schedule, and performance. Cash outflows include expenditures for labour, equipment, and materials, and cash inflows take the form of progress payments for completed work and a final payment paid upon completion of the entire project [7]. However, as PSP often consists of hundreds of activities, numerous authors have developed exact and heuristic methods for the resource-constrained problem (RCPSP) where a schedule is derived by allocating limited resources to competing activities so that a project's objective is achieved [4, 8-10]. The investment analysis tool that considers the time value of money is the net present value (NPV) analysis [3]. Net present value (NPV) is an effective measure of project financial performance because it balances the objectives of minimising project makespan and maximising project value [11]. Some people involved in short projects, which are typically targeted for 6 to 12 months for implementation, might think that the time value of money is not an important issue. However, the financial analysis of projects should be based on the useful life of the project not the implementation time [11]. It is vital to apply NPV analysis even to short projects [3].

1.2 Project scheduling problem (PSP)

1.2.1 Type of project scheduling problem

Over the last decade, scheduling problems have been studied intensively in the literature. The project scheduling problem can be categorised into two types; P-Problems and NP-hard Problems as illustrated in Figure 1.1

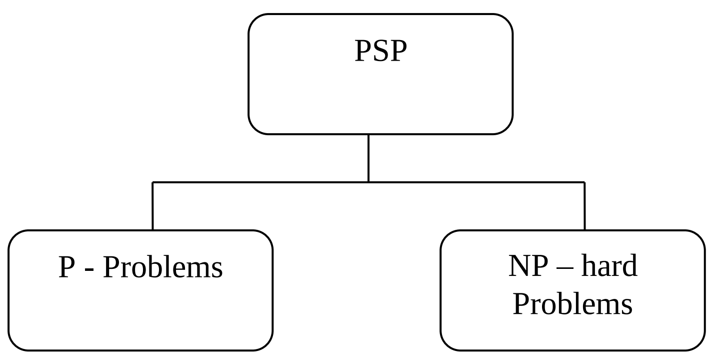


Figure 1.1 Types of project scheduling problem modified from [11]

P-Problems (P) are problems, which there are existed optimal solution algorithms of polynomial complexity. It can be solved with the respective optimal algorithm and the research is focused on the complexity reduction [11].

NP-hard problems are the problems that can be solved only by non-polynomial complexity algorithms. The RCPSP is a generalisation of the static job shop problem and hence belongs to the class of NP-hard problems [2]. From [11], there are four directions of attacking with NP-hard problems.

(i) *Relaxation.* One can solve a problem close to the original one, making weak some parameter or allowing pre-emptions [12].

(ii) *Approximate optimisation.* The problem is solved with the aid of some heuristics, whereas the search is directed towards the worst-case or mean performance analysis trying to construct a more effective heuristic [13].

(iii) *Enumerative optimisation.* This approach is followed only if the objective is to find the truly optimal solution. The implication of this approach is the increased (usually exponential) complexity of the algorithms, even if they are pseudopolynomial. The techniques used for this type of optimisation include dynamic programming, branch and bound, iterative techniques, etc [14].

(iv) *Expert systems.* This approach is currently finding increased use for the solution of NP-hard scheduling problems [15]. It usually help in the effective combination of more than one heuristic [16].

1.2.2 Objective of project scheduling

Makespan minimisation is probably the most widely applied objective in the project scheduling domain [2]. The makespan is defined as the time span between the start and the end of the project. Since the start of the project is usually assumed to be at t = 0, minimising the makespan reduces to minimising the maximum of the finish times of all activities. Makespan minimisation is a regular performance measure.

Net present value (NPV) maximisation: When significant levels of cash flows are presented in the project, in the form of expenses for initiating activities and progress payments for completion of parts of the project, the NPV criterion is an appropriate measure of project performance [3]. This criterion generates a cost-critical path and schedule of activities, in contrast to the time-critical path and schedule obtained by the makespan objective. Much of the research on the NPV project scheduling problem has concentrated on designing solution approaches for the RCPSP with cash flows, where the problem is to maximise the NPV of the project subject to precedence and renewable resource constraints [17]. The solution to the mathematical models provides both the optimal-scheduled start time of each activity as well as the optimal project NPV.

1.2.3 Capital budgeting

Capital expenditures (CAPEX or capex) are expenditures creating future benefits. A capital expenditure is incurred when a business spends money either to buy fixed assets or to add to the value of an existing fixed asset with a useful life extending beyond the taxable year [17].

An operating expense, operating expenditure, operational expense, operational expenditure or OPEX is an ongoing cost of running a product, business, or system [18]. In contrast, a capital expenditure (CAPEX), is the cost of developing or providing non-consumable parts for the product or system [19].

Project cash flow

On the statement, cash flows are segregated based on source [20]:

- *Operating activities*: involve the cash effects of transactions that enter into the determination of net income [18].
- *Investing activities*: concern with buying (and selling) property, plants, and equipment (PPE); acquiring and disposing of securities of other entities [21];
- *Financing activities*: include issuance and reacquisition of a firm's debt and capital stock, and dividend payments [22].

Project net cash flow = Gross revenue – expenditure

= Gross revenue - CAPEX - OPEX - royalty – tax [23].

A typical project cash flow is shown in Figure 1.2, along with a cumulative net cash flow showing how the cumulative revenue is typically split between the CAPEX, OPEX [23]. The cumulative amount of money occurring to the company at the end of the project is the cumulative net cash flow (Figure 1.3).

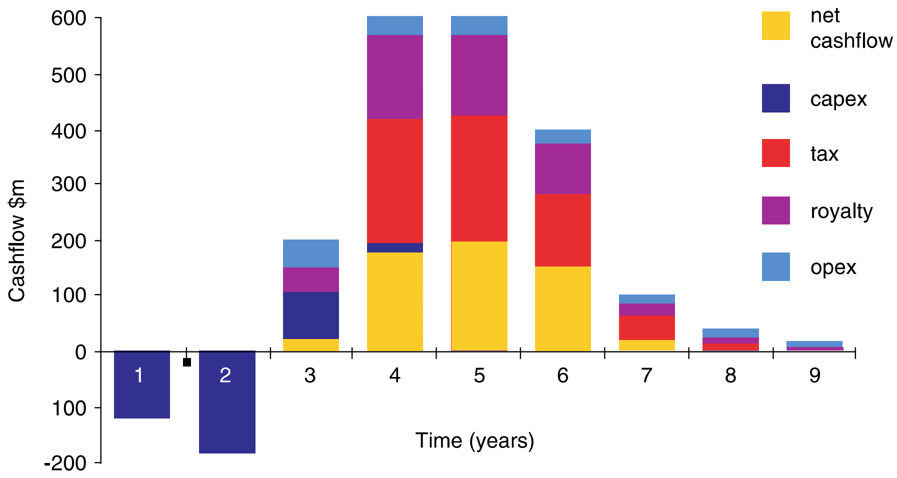


Figure 1.2 Components of a project cash flow [20].

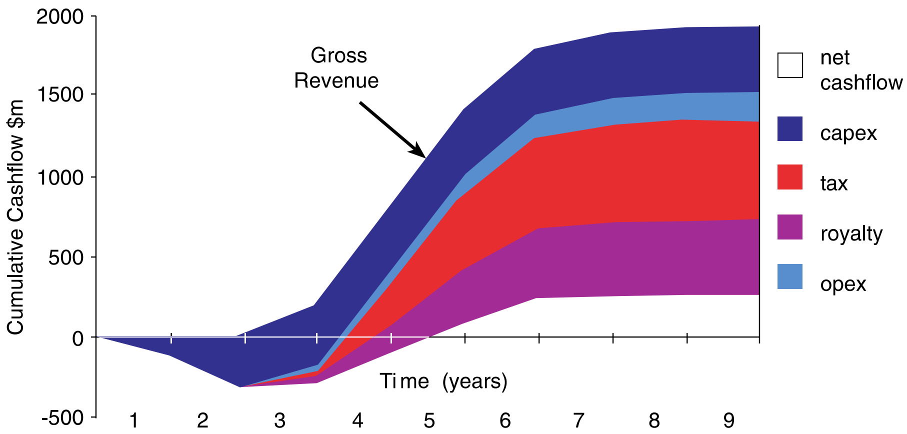


Figure 1.3 The cumulative cash flow [20].

The net cash flow determines the economic lifetime of the field. From Figure 1.4, the most negative point on the cumulative net cash flow indicates the maximum cash exposure of the project. If the project were to be abandoned at this point, this is the greatest amount of money that the investor stands to lose, before taking account of specific contractual circumstances (such as penalties from customers, partner claims, contractors' claims) [23]. It also represents the funds, which are required to finance the project – if the maximum exposure is greater than the company's capacity to raise the capital then the investor may consider farming out a portion of the project to a joint investor.

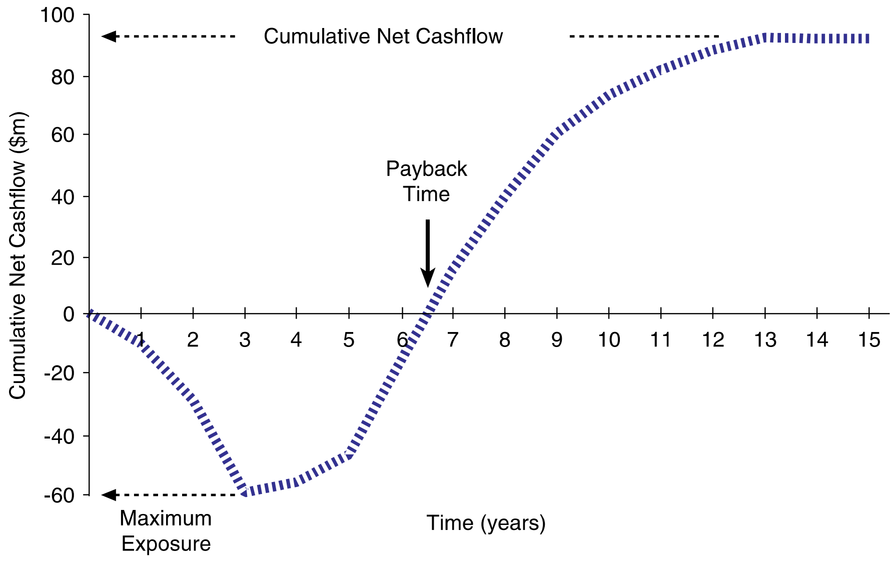


Figure 1.4 Indicators from the cumulative net cash flow [20].

The point, at which the cumulative net cash flow turns positive, indicates the payback time. This is the length of time required to receive accumulated net revenues equal to the investment. Payback time is primarily an indicator of risk – the longer the payback the more risky the project, but it states nothing about the net cash flow after the payback time and does not consider the total profitability of the investment opportunity. Payback time indicates how long it will take to get the investment funds back. The cumulative net cash flow accrues to the investor at the end of the economic lifetime of the project.

Discounted Cash Flow

The annual net cash flows now need to incorporate the timing of the cash flows, to account for the effect of the time value of money. The technique which allows the values of sums of money spent at different times to be consistently compared is called discounting [24].

Capital Budgeting techniques

The ASQ's Six Sigma Body of Knowledge [25] suggests several financial metrics that can be used to assess the economic impact of a project and hence provide guidance in the project selection process:

- Payback period (PP)
- Return on assets (ROA)
- Return on investment (ROI)
- Internal Rate of Return (IRR)
- Net present value (NPV)
- Profitability index (PI)

Payback period: PP is the length of time necessary for net cash benefits or inflows from a project to equal the net costs or outflows of the project [26]. The project with the shortest payback period gets funded first, then the next shortest, until the resources are exhausted. This metric ignores the time value of money (i.e. the value of a dollar today is not the same as its value tomorrow) [6].

Return on assets: ROA is net income divided by total assets, where net income for a project is the expected earnings, and total assets is the value of the assets applied to the project [26]. Projects with the largest ROA are approved first. This metric does not take into consideration the time value of money.

Return on investment: ROI is net income divided by investment, where net income for a project is the expected earnings, and investment is the value of investment in the project [27]. Projects with the largest ROI are approved first. This metric also ignores the time value of money [6].

Internal Rate of Return: IRR is defined as the discount rate that makes the NPV equal to zero. IRR rule is to accept a project if the IRR is higher than the opportunity cost of capital [28].

The Net present value: NPV is the sum of the ‘‘real’’ cash flows from a project over time. Projects with the largest NPV are approved first. NPV analysis does consider the time value of money as it provides results in deflated dollars (or as economists say, ‘‘real’’ or ‘‘current’’ dollars)” [2].

Profitability index: PI is a ratio of the present value of the benefits (BPV) to the present value of the costs (CPV).

$$PI = \frac{The\ cumulative\ net\ cash\ flow}{Total\ capital\ expenditure}$$

$$PI = \frac{BPV}{CPV}$$

Three decision criteria methods— the net present value (NPV), the internal rate of return (IRR), and profitability index (or the *benefit-cost ratio*), can be properly applied to the design project acceptance problem [29]. This is particularly the case with estimates of *NPV* and *IRR* have estimated the life-cycle costs during the engineering design stage. From [3], these criteria are the so-called *rational criteria* because they take into account the two attributes most often absent in other criteria:

- The entire cash flow for the life of the project
- The time value of money.

The Net Present Value method and its criterion

The vast majority of the projects scheduling methodologies presented in the literature have been developed with the objective of minimising the project duration subject to various types of precedence and resource constraints. In doing so, the financial aspects of project management are ignored [30]. If financial aspects are taken into consideration of particular interest to project management, the maximisation of project NPV then is decided as the more appropriate objective [31]. Project scheduling problems arise where the NPV of the project is to be maximised.

Net Present Value (NPV) is one of Economic evaluation methods. The idea of maximising the NPV of the cash flows of a project as a concise and financially relevant criterion in deciding on the timing of activities in a project was introduced many years ago [32].

Let BPV_x be the present value of the benefit of a project x, CPV_x be the present value of the cost of the project x, and a MARR be a minimum attractive rate of return Then, for the MARR = r over a planning horizon of n years,

Equation (1.1) The present value of benefits of a project [2]

$$\mathrm{BPV_x} = \sum_{t=0}^{n} B_{t,x}(1+r)^{-t} = \sum_{t=0}^{n} B_{t,x}(P|\mathrm{F},\mathrm{r},\mathrm{t}) \qquad (1.1)$$

Equation (1.2) The present value of costs of a project [2]

$$\mathrm{CPV_x} = \sum_{t=0}^{n} C_{t,x}(1+r)^{-t} = \sum_{t=0}^{n} C_{t,x}(P|\mathrm{F,r,t}) \qquad (1.2)$$

where the symbol (P|F,r,t) is a discount factor equal to $(1+r)^{-t}$ and reads as follows: "To find the present value P, given the future value F=1, discounted at an annual discount rate r over a period of t years". The present value is obtained when the benefit or cost in year t is multiplied by this factor. The NPV of the project x is calculated as:

Equation (1.3) The Net Present value of a project [2]

$$\mathrm{NPV_x} = \sum_{t=0}^{n} BPV_x - CPV_x \qquad (1.3)$$

Or

Equation (1.4) The Net Present value of a project [2]

$$\mathrm{NPV_x} = \sum_{t=0}^{n} (B_{t,x} - C_{t,x}(P|\mathrm{F,i,t}) = \sum_{t=0}^{n} A_{t,x}(P|\mathrm{F,i,t}) \qquad (1.4)$$

If there is no budget constraint, then all independent projects having NPV greater than or equal to zero are acceptable. The project x is acceptable as long as the following equation shown.

Equation (1.5) NPV of acceptable project [31]

$$\mathrm{NPV_x} \geq 0 \qquad (1.5)$$

For mutually exclusive proposals (x = 1, 2,... ,m), a proposal j should be selected if it has the maximum nonnegative NPV among all m proposals.

Equation (1.6) Proposal for selecting project NPV [31]

$$NPV_j = max_{x \in m}(NPV_x) \quad (1.6)$$

provided that $NPV_j \geq 0$.

This section is to examine the NPV criterion.

The NPV concept lies at the very heart of capital budgeting and finance [32]. Wise investment decisions are supposed to be based on a very simple principle.

- The value of an amount of money is a function of the time of receipt or disbursement of cash [33].
- A dollar received today is more valuable than a dollar to be received in some future time period, because the dollar today can be invested to start earning interest immediately.

The *accept-reject decision* of an independent project is then the result of a very simple mechanism.

1. Choose an appropriate discount rate r (also called the hurdle rate or the opportunity cost of capital), representing the return foregone by investing in the project rather than investing in securities.

The discount factor $\beta = (1 + r)^{-1}$

= Denotes the present value of a dollar to be received at the end of period 1 using a discount rate r [31].

2. Estimate the future incremental cash flows on an after tax basis and compute the NPV of the project using the formula:

Equation (1.7) NPV of a project (The Cash flow formula) [32]

$$\mathrm{NPV} = \mathrm{CF}_0 + \sum_{t=1}^{\infty} \frac{CF_t}{(1+r)^t} \qquad (1.7)$$

Where

CF_0 = cash flow (usually a negative number representing the initial investment outlays) at the end of period 0 (that is, today)

CF_t = cash flow at the end of period t.

Sometimes, Eq. (1.7) is replaced by its continuous equivalent assuming continuous discounting. The discount factor is then simply replaced by exp(-α).

3. The rule is then to accept the project if the NPV is greater than or equal to zero and to reject it when the NPV is less than zero.

Since it seems viable to assume that most project contractors have their primary goal as the maximisation of their returns, not the least their financial returns [34], the expanding literature on project scheduling with discounted cash flows takes the fundamental view that it is appropriate not only to base the accept-reject decision but also to schedule projects in order to accomplish some optimisation of financial returns [32].

In recent years, a number of publications have dealt with the project scheduling problem under the NPV objective. Research efforts have led to optimal procedures for the unconstrained project scheduling problem, where activities are only subject to precedence constraints [35].

1.3 Current scheduling methods

The two methods for network drawing, activity-on- arrow and activity-on-node or precedence diagram, are both in widespread use. Although the construction industry seems to favour precedence diagrams, arrow diagrams still appear to be the most popular overall [36]. Similarly, two approaches have been adopted for time analysing projects, namely, Critical Path Analysis (CPA) (characterised by one time estimate per activity) and Program Evaluation and Review Technique (PERT) (characterised by the three-time estimates per activity) [37].

CPM [36] is a deterministic technique that, by using a network of dependencies between tasks and given deterministic values for task durations, calculates the longest path in the network called the 'critical path'. The length of the 'Critical Path' is the earliest time for project completion. The critical path can be identified by determining the following parameters for each activity: *D* – Duration, *EST* - earliest start time, *EFT* - earliest finish time, *LFT* - latest finish time and *LST* - latest start time.

The earliest start and finish times of each activity are determined by working forward through the network and determining the earliest time at which an activity can be started and finished considering its predecessor activities [38]. For each activity j:

Equation (1.8) Earliest Start Time Function [36]

$$EST_j = \text{Max}\ [EST_i + D_i\ ;\ \text{over predecessor activities } i] \qquad (1.8)$$

Equation (1.9) Earliest Finish Time Function [36]

$$EFT_j = EST_j + D_j \qquad (1.9)$$

The latest start and finish times are the latest times that an activity can be started and finished without delaying the project and are found by working backward through the network [38]. For each activity i:

Equation (1.10) Latest Finish Time Function [36]

$$LFT_i = \text{Min}\ [LFT_j - D_j\ ;\ \text{over successor activities } j] \qquad (1.10)$$

Equation (1.11) Latest Start Time Function [36]

$$LST_i = LFT_i - D_i \qquad (1.11)$$

Activity's 'Total Float' (TF) (i.e. the amount that activity's duration can be increased without increasing the overall project completion time) is the difference in the latest and earliest finish of each activity [34]. A critical activity is the one with no TF and should receive special attention (delay in a critical activity will delay the whole project). The critical path then is the path(s) through the network whose activities have minimal TF [39].

The CPM approach is uncomplicated and it provides very useful and fundamental information about a project and its activities' schedule. However, it is too simplistic to be used in real complex projects due to its' single point estimate assumption [40]. The challenge is to incorporate the inevitable uncertainty [41]. Construction professionals are heavy users of the CPM techniques assisted by project management software [42] [37]. Microsoft Project™ is one of the most widely-used systems in project planning and control [43]. Those software tools have assisted project managers in the planning and administrating of complex projects, performing large, and repetitive calculation [19, 42].

However, a number of techniques such as *Program Evaluation and Review Technique* (*PERT*), *Critical Chain Scheduling* (CCS) and *Monte Carlo Simulation* (MCS) do as follows:

PERT [36, 44, 45] incorporates uncertainty in a restricted sense, by using a probability distribution for each task. Instead of having a single deterministic value, three different estimates (pessimistic, optimistic and most likely) are approximated. The critical path and the start and finish date are calculated by the use of distributions' means and applying probability rules. Results in PERT are more realistic than CPM [46].

Critical Chain (CC) Scheduling is based on Goldratt's Theory of Constraints (TOC) [47]. For minimising the impact of Parkinson's Law (jobs expand to fill the allocated time), CC uses a 50% confidence interval for each task in project scheduling. The safety time (remaining 50%) associated with each task is shifted to the end of the critical chain (the longest chain) to form the project buffer [48]. Although it is claimed that the CC approach is the most important breakthrough in project management history, its over-simplicity is a concern for many companies who do not understand both the strength and weakness of CC and apply it regardless of their particular and unique circumstances [49]. The assumption that all task durations are overestimated by a certain factor is questionable and the main issue is: "*How does the project manager determine the safety time?*" [48]. CC relies on a fixed, right-skewed probability for activities, that may be inappropriate[50] and a sound estimation of project and activity duration (and consequently the buffer size) is still essential [51].

Monte Carlo Simulation (MCS) was first proposed for project scheduling in the early 1960s [52] and implemented in the 1980s [53]. In the 1990s because of improvements in computer technology, MCS rapidly became the dominant technique for handling uncertainty in project scheduling [54]. A survey by the Project Management Institute [PMI 1999] [52] showed that nearly 20% of project management software packages support MCS. For example, PertMaster accepts scheduling data from tools like MS-Project and Primavera and incorporates MCS to provide project risk analysis in time and cost [55]. However, the Monte Carlo approach has received some criticism. Van Dorp and Duffey [41] explained the weakness of Monte Carlo simulation, in assuming statistical independence of activity duration in a project network. Moreover, being event-oriented (assuming project risks as 'independent events'), MCS and its implement tools do not identify the sources of uncertainty [56].

Undoubtedly, CPA is the more favoured in practice and many debatable assumptions associated with PERT have assured its limited practical use [56]. PERT addresses projects where the probabilistic element of activity durations is an important factor. Monte Carlo Simulation has been adopted in practice as a more suitable method of network time analysis [57].

In recent years, a number of publications have dealt with the project scheduling problem under the NPV objective [58]. Research efforts have led to optimal procedures for the unconstrained project scheduling problem, where activities are only subject to precedence constraints. In addition, numerous efforts aim at providing optimal or suboptimal solutions to the project scheduling problem under various types of constraints (capital constrained, different resource types, different materials, time/cost trade-offs) [59] [60].

1.4 Heuristics approaches

Project scheduling problem (PSP) is solved by two distinctly different approaches. The first includes optmisation approaches that produce optimal solutions. Although these techniques produce optimal solutions, they fail to solve the relatively medium-size and more complicated problems usually encountered in practice. The second approach includes heuristic methods. The idea behind heuristic algorithms for constrained project scheduling is to rank the activities by some rule [5]. This may be managerial priority, earliest start times, most resource 'greedy' or any other project related value, and to schedule the activities in that ranking order ensuring that the constraints on the project are never exceeded. Thus, activities considered to be ‘important' in some sense are scheduled as soon as possible.

There are of course many heuristic algorithms available in practice and though they can be tested relative to each other on a particular project it has not yet been possible to classify constrained projects such that a suitable heuristic may be selected. Gonguet [16] describes an early attempt at comparing heuristics. Davis [4, 61] suggested the approach of 'try as many heuristics as you can in the time available'. Davis and Patterson [9, 62] and Herroelen [62] comparing optimum seeking and heuristic algorithms. The demand however for 'good' heuristics is at least matched by the practical requirement for them to be embodied in user-friendly computer programmes allowing a full range of resource facilities to be modelled.

So far, most published constrained project scheduling algorithms have focussed on algorithm quality using project duration as the only criterion [5]. They have assumed fixed activity durations, fixed resource requirements over activity durations and fixed resource limits over time. More detail on heuristics approached will be revealed and discussed further on Chapter 3 and 4.

1.5 Aim of the research

The aim of this research is to propose a heuristics rule that can achieve the objective in the form of optimising the project NPV.

In order to achieve this aim, there is a sequence of objectives, which have to be succeeded.

1.6 Objectives of the research

1. To investigate the approaches which have been developed for solving the project scheduling problems and cash flow management.
2. To examine the significant relationship between each capital budgeting technique.
3. To evaluate the performance of existing scheduling rules and techniques.
4. To consider an alternative heuristic scheduling technique with improves performance.
5. To apply the model to a wide range of different projects.
6. Propose the new improve heuristic scheduling technique to minimise the capital expenditure.

1.7 Research questions

(i) Is NPV the most effective measure for evaluating project investment?

(ii) Can the existing priority rule based heuristics scheduling techniques be improved?

(iii) Is it possible to optimise the NPV of the project subject to a late start scheme?

(iv) Does project complexity affect the efficiency of the priority rule based scheduling techniques?

1.8 Research scope and limitation

The objective of this study is to find the feasible schedule for all activities such that the NPV of the project is maximised. The investment analysis tool that does consider the time value of money is NPV analysis [63]. The investment analysis tools such as PP (Payback Period) and IRR do not take into consideration the time value of money and also consider the effect of inflation, which can have a significant impact on the results of the analysis [63]. Some people involved in short projects, which are typically targeted for 6 to 12 months for implementation, might think that the time value of money is not an important issue. However, this would be an erroneous conclusion because, even though the implementation time may be short, the useful project life can be much longer [64]. The financial analysis of projects should be based on the useful life or a long-term view of the project.

The problem in this research is referring to constrained project scheduling problems in deterministic environment. A project begins with a fixed amount of capital during the construction phase. A series of cash flows occur over the course of a project in two forms. Cash outflows include expenditures for labour, equipment, and materials. Cash inflows take place in the form of progress payments for completed work, and may be added to the capital balance available for reinvestment in the project.

The activities are interrelated by constraints: Precedence constraints - as known from traditional CPM-analysis - force an activity not to be started before all its predecessors have been finished; and the capital-constrained where investment in project activities is constrained by a capital constraint. Each activity is assumed to have known the duration and the activity once started cannot be interrupted.

The following assumptions are made in this study:

- The precedence relationships among activities are deterministic. Each activity cannot start until its predecessor activities have finished.
- The duration of each activity is known and fixed. The quantity of capital available in each period is known and remains constant. Any remaining resources at the end of each period cannot be used in any later period.
- Once an activity is started, it cannot be interrupted. Also, preemption is not allowed.
- The cash flow for each activity is known. The discount rate is also known.

<u>Discount rate</u> - In business investment opportunities, the appropriate discount rate is the cost of capital to the company. This may be calculated in different ways, but it should always reflect how much it costs the company to borrow the money, which it uses to invest in its projects. This may be a weighted average of the cost of the share capital and loan capital of a company.

If the company is fully self-financing for its new ventures, then the appropriate discount rate would be the rate of return of the alternative investment opportunities (i.e. other projects) since this opportunity is foregone by undertaking the proposed project. This represents the opportunity cost of the capital. It is assumed that the return from the alternative projects is at least equal to the cost of capital to the company, otherwise the alternative projects should not be undertaken. The appropriate discount rates would be 0% (undiscounted), 10% (the cost of capital), and 20% (the cost of capital plus an allowance for risk) [2]. This study uses 20% as a discount rate on first and second phase experiment. The third phase use the actual discount rate from each project. More details and assumptions will be discussed further in this thesis.

1.9 Publications and conferences

The author has produced three peer reviewed conference papers and presented at two international conferences in Singapore and UK. One oral presentation has been presented at international conferences organised by the University of Salford.

A journal paper is in the process of being written and will be submitted to the International Journal of Project Management. A full list of publication citations is available in Appendix J.

1.10 Thesis structure

- Chapter 1 Introduction

This chapter provides the background to the research, sets out its aims and objectives, research questions, literature review, research scope and limitation and structure of the thesis.

- Chapter 2 Research Methodology

This chapter provides a prelude to the methodology of the research. The research philosophy, approach and strategy are discussed. Types of methodologies to be used in this study are identified along with the research process. The data collection and analysis methods are also identified.

- Chapter 3 Literature Review I: Scheduling techniques

This chapter presents literature review on the two aspects of this research, that is the critical review of capital budgeting techniques, and an in depth study of the Project scheduling methods and techniques. In order to understand this concept, discussion on terms complex and complicated has been presented, along with the underlying concepts used by different researchers to explain project scheduling techniques.

- Chapter 4 Literature review II: Maximise NPV

This chapter presents the literature review focusing on scheduling techniques with max NPV. The views of various researchers on each scheduling techniques and their applicability and usefulness have been presented.

- Chapter 5 [m-CCF] heuristic algorithm

This chapter provides an overview, background and procedure of the proposed technique used in this study. The detail description of each techniques and flow chart are also included.

- Chapter 6 Results I: A review of algorithm performance

This chapter presents the analysis and findings of the first phase experiment. The primary aim of these experiments was to get an initial exploratory view on each scheduling techniques, and compare this with the theoretical concepts.

- Chapter 7 Results II: [m-CCF] Validation

This chapter details the analysis, results, and findings after the first phase experiment. The purpose of this experiment was to assess the performance of the propose rule, m-CCF and to validate the findings of the previous studies.

- Chapter 8 Discussions and Recommendations

This chapter provides an analysis and discussion of each technique in order to achieve the research objectives and to address the research questions. Each element of the research investigations and findings from each case is compared and summarised.

- Chapter 9 Conclusions and Future work

This chapter provides a summary and conclusions to the whole research study. Implications of the study will also be discussed and suggestions for future research will be presented.

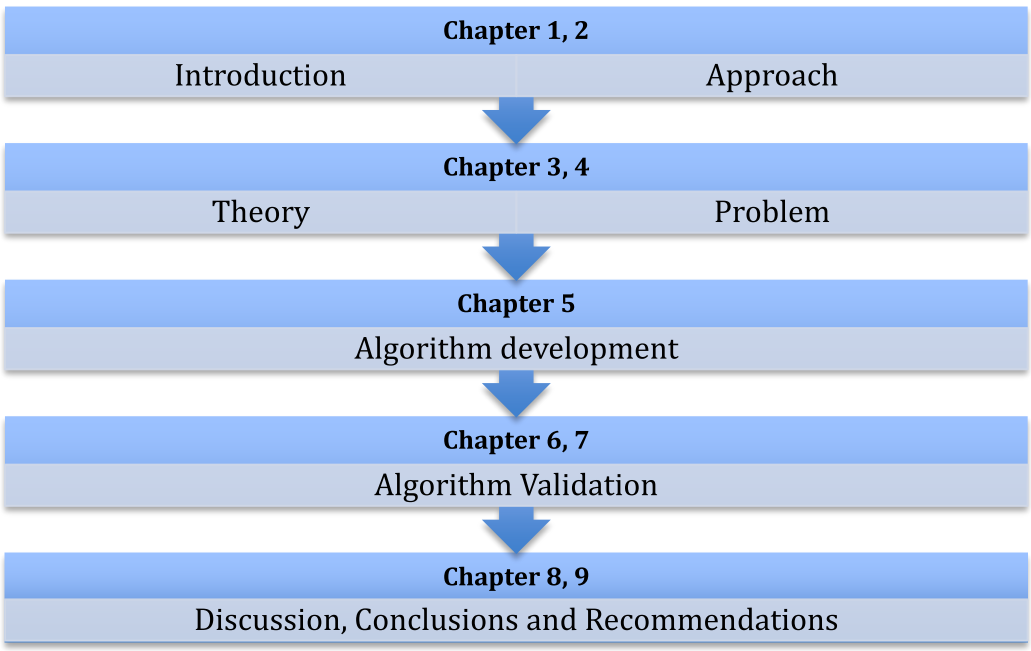


Figure 1.5 Thesis structure

Chapter 2: Research Methodology

2.1 Introduction

This chapter discusses the methodology adopted for this project, the research philosophy is stated, the research approach are defined, and the research strategies used in this research for answering the research questions is described. The methods used to gather data and analysis are defined and described.

2.2 Research Philosophy

An understanding of research philosophy and methodology is essential so that the most appropriate design and methods can be applied to address the research objectives.

Easterby-Smith et al. [65] gave three reasons why understanding research philosophy is useful:

1. *"It can help clarify research designs, including how the evidence is gathered, interpreted and how this will provide answers to the research questions.*
2. *It will help clarify the limitations of specific approaches.*
3. *It can help the researcher understand designs outside their experience and adapt these according to the constraints of the subject."*

Bryman [66] classified ways of thinking to two main philosophies: ontological and epistemological as shown below.

"Epistemology: Assumptions about the best ways of enquiring into the nature of the world

Ontology: Assumptions about the nature of reality"

The ontological philosophy involves the logical investigation of the different ways in which the different types of things are thought to exist, and the nature of the various kinds of existences.

The epistemological philosophy addresses the question of what is regarded as acceptable knowledge in a discipline. The question of whether the social world can and should be studied according to the principles, procedures and ethos as the natural sciences is the central issue of this philosophy. Bryman [66] further divided epistemological philosophy into positivist and interpretivist approaches.

The interpretivist approach is related to knowledge development and theory built through developing ideas inducted from the observed and interpreted social constructions (qualitative approach), whereas the positivist approach is associated with knowledge development by investigating the social reality through observing objective facts (quantitative approach).

Saunders [67] classified research into six stages and labelled the model which presented them as 'the research onion'. This whole research process is captured in the below in Figure 2.1. Saunders divided the research to include: philosophies; approaches; strategies; choices; time horizons; techniques and procedures.

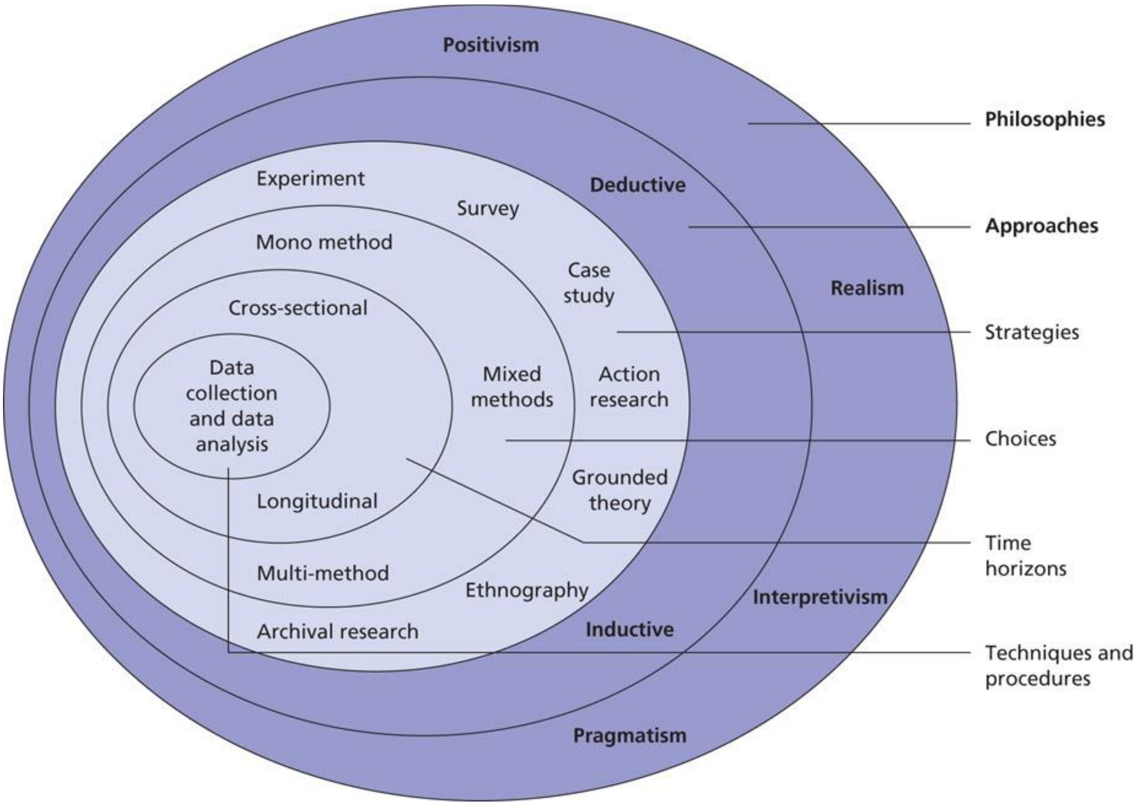


Figure 2.1 Research onion [67]

In the existing literature, positivism and phenomenology appear to be the research paradigms that are applied to explore the truth and facts about the world by researchers. These two stances dominate epistemology. The alternative terms used for these two terminologies are shown below [68].

- Positivist paradigm**:** Quantitative, Objectivist, Scientific, Experimentalist, Traditionalist, Hypothetical deductive, Social constructionism.

- Phenomenological paradigm: Qualitative, Subjectivist, Humanistic, Interpretivist / hermeneutic, Inductive

2.1.1 Positivism

The research philosophy of positivism is to develop a strategy and gather these data from existing literature about the general topic for describing the causal relationships to build hypotheses that already been tested [67]. As a result of existing studies, it leads to further investigate. According to the attribute of positivism, it is compatible for researcher to build a model and hypotheses from existing studies in order to avoid the risk of error of each constructs and fast method to gain general knowledge in specific topic.

The positivism method gives consequences with a wide range of phenomenon with a large relevant sample size. Eventually, it is quick and economical techniques [69]. However, the result cannot be efficient for deeply comprehending the people feeling behind their actions. Therefore, it is hard to build new theory, predict and explain the change of these actions in the future [69].

2.2.2 Realism

The philosophy of realism represents the real procedure that existing independently of human mind within different social conditions. The procedure of developing hypotheses and construct is similar to positivism [70]. On the contrary, during the data collection and interpretation, the better understanding of each social element is essential [71]. Thus, for the data collection and interpretation, the realist philosophy will be applied for different country context. Bhaskar [72] suggested that researcher will better understand the change and trend of specific phenomenon better in the social world and social context because researcher who used the realism will only see and understand the phenomenon in one piece (one country) of bigger picture (world). However, the misinterpretation and inadequate information can easily occur due to the cultural bias of researchers [71].

2.2.3 Interpretivism

The philosophy of interpretivism is referred to the understanding of multiple reality and motivation behind the action with divergent social context surrounding among people, which is opposed to positivism [71]. Amaratunga et al. [69] pointed out that researcher has ability to understand the change of phenomenon and people's behavior and mind. However, the confusion for researcher to understand the world phenomenon can be occurred from much point of views. In consequence, it is difficult for researcher to neutral for data collection and interpretation.

2.2.4 Pragmatism

Pragmatist philosophy illustrates the multiple methods for better answering research question. Tashakkori and Teddlie [73] suggested that this method is appropriate with the specific study when it comes to search information for better understand the knowledge and data. They exclaimed that pragmatism helps researcher to get rid of irrelevant information and concepts as the fundamental of other methods to adapt. Researchers use multiple methods for gaining knowledge and different prospects for data interpretation [67].

According to Norman et al [74], there are overlapping meaning and boundary between paradigms, which causes confusion and serious issue of incommensurability.

2.3 Research Approaches

Authors have used different expressions to define the research approaches, and irrespective of the notion used, these research approaches use a variety of research methods and techniques for data collection [75]. For the empirical approach, the main dimensions considered are,

- Qualitative / Quantitative
- Deductive / Inductive

2.3.1 Quantitative/Qualitative Approach

Qualitative research is defined as, *'a subjective approach which includes examining and reflecting on perceptions in order to gain understanding of social and human activities'* (Hussey and Hussey [76]).

Qualitative approach is often adapted when it is required to uncover a person's experience or behaviour, to create an in-depth analysis of a particular process of a single case study or limited number of cases, and to understand a phenomenon about which little is known [77]. Qualitative data sources include interviews, questionnaires and surveys (open-ended), documents and texts, observations (field work), focus groups, and researcher's impressions and reactions to understand and explain the social phenomenon [78].

The motivation for doing qualitative research, as opposed to quantitative research, comes from the observation that, if there is one thing, which distinguishes humans from the natural world, it is their ability to talk. Qualitative research methods are designed to help researchers to understand people and the social and cultural contexts within which they live [79].

Quantitative research is more *objective* in nature than the qualitative research, and the emphasis of quantitative research is on collecting and analysing numerical data; as it concentrates on measuring such as the scale, range, frequency of a phenomenon [69]. This type of research, although initially harder to design, is usually highly detailed and structured, and results can be easily collated and presented statistically. Quantitative research methods were originally developed in the natural sciences to study natural phenomena.

2.3.2 Deductive/Inductive

Deductive approach is one in which a theory and hypotheses are developed and then a strategy is designed to test the hypotheses, whereas in the inductive approach data is collected and theory is developed as the result of the data analysis [67]. Deductive approach works from the more general to the more specific, informally called a "top-down" approach, beginning with a theory, narrowing down into specific hypotheses and finally testing them [76], as shown in Figure 2.2.

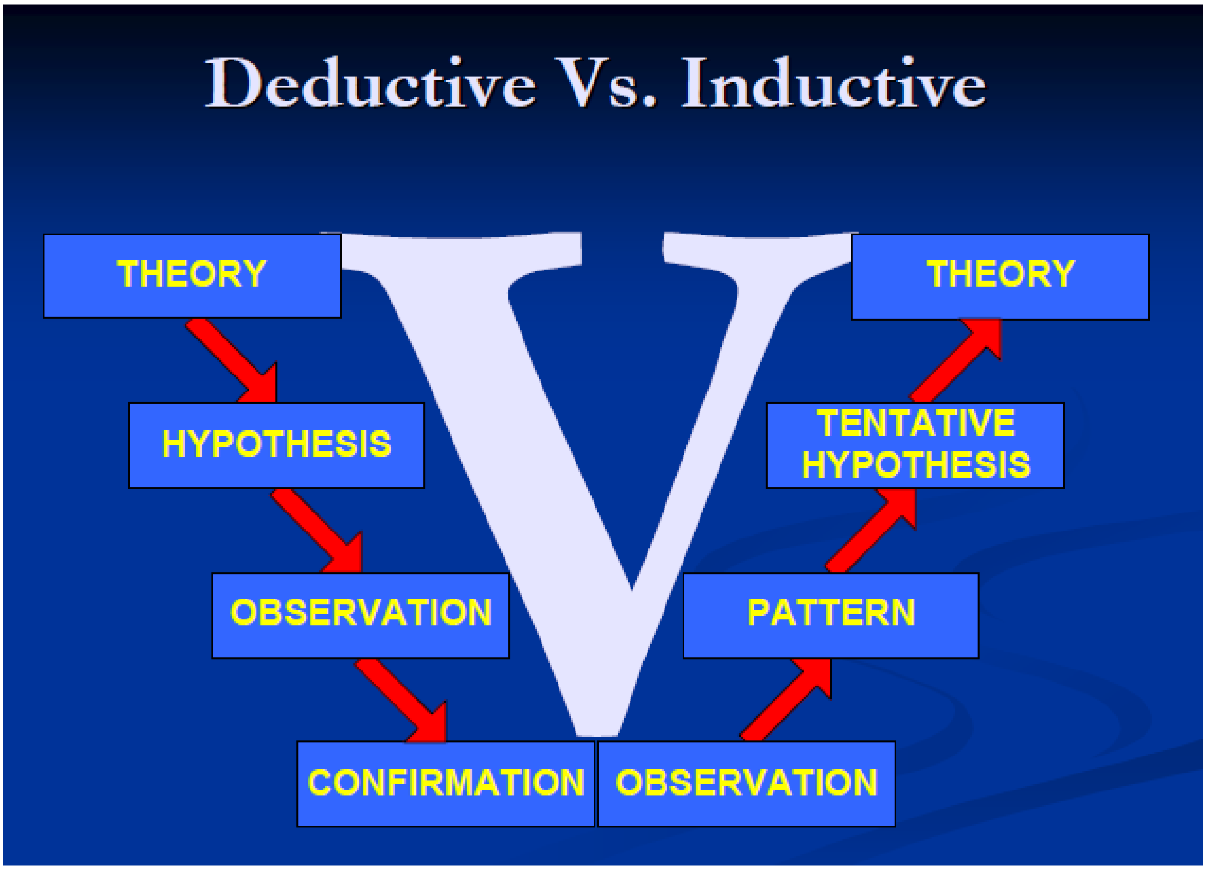


Figure 2.2 Deductive and Inductive Approach. Adapted from Burney [80]

Inductive approach works the other way, moving from specific observations to broader generalisations and theories, informally called a “bottom up” approach, beginning with specific observations and measures, detecting patterns and regularities, formulating some tentative hypotheses to be explored, and finally ending up developing some general conclusions or theories.

2.4 Research Strategy

The various strategies have been highlighted along with a brief description of each. The following sections will describe the most commonly used research approaches.

2.4.1 Experiment

Experiment is concerned about original relationship among another dependent variable that are affected from one independent variable that can be measured [81]. Experiment is utilised to illustrate and investigate for answering the question “how” and “why” [67]. Researcher has a high control over the circumstances of research procedure [82]. Consequently, the causal relationship and strict research design allows experiment strategy to be the strongest method for experiment the relationship among variables [83]. However, experiment is the high cost for a large scale with long period of time in order to maintain the credibility and high control of the experiment [83]. Hakim [81] suggested that experiment could be more complicated as majority of sample size is small which leads to external validity issues.

2.4.2 Survey

Survey is related to deductive approach with its popular strategy to response who, what, where, how much and how many questions [67]. Survey is associated with large sample size with a series of questions. The survey strategy gives ease, reliability and simplicity to researchers with a fixed response of multiple-choice questionnaire which is already standardised, coded from previous studies in order to control and improve internal validity for its simplification of coding, analysis and interpretation of data. Malhotra [84] suggested that survey is highly economical and more control over the research procedure. However, there are some limited questions for respondents who are unwilling to answer and provide information [85].

2.4.3 Case study

Case study particularly answers why, what and how questions with its purpose to study in specific and unique case of real life context [67]. Researchers can use multiple and various case studies and sources in order to get rid of bias result that is called triangulation [78]. However, due to its limited ability of variables and data, it is hard for researchers to better comprehend the study. Because of case study is limited to answer what and how questions, it is often used as a complimentary with other [67].

2.5 Time-horizons

2.5.1 Cross-sectional studies

Cross-sectional designs are sometime called sample survey because it is a one particular point of time for the study [84]. The data is collected and analysed only once. Thus, researcher adopts the study with cross-sectional designs owing to its attributes of short time period. It is common for academic student to apply this method because of time constraint.

According to Saunders, Lewis and Thornhill [71], this type of method is economical method to choose the sample with well represented of entire population with well interest characteristic with the study and short time consuming. However, it is difficult to predict the phenomenon in the future as the study cannot detect change of the phenomenon and the data can be considered as inaccuracy.

2.5.2 Longitudinal studies

Longitudinal designs are associated with watching people over time, which allows researcher to highly control over variables. Researchers can measure same variables at different point of time for observing the change [70]. Researchers can investigate the change and development of the phenomenon with repeat measurement and the data is accurate. However, it involves large amount of data because researchers have to collect data in different time for watching the change. According to Malhotra [84] , the sample will not be representative of population of interest, which caused from the high refusal and drop out respondents.

2.6 Optimisation Methods

As this study applies optimisation routine to solve scheduling problems, this section presents a list of problem types, arranged in order of increasing difficulty for the solution methods [86].

- Linear Programming (LP) Problems
- Nonlinear Programming (NLP) Problems

- Quadratic Programming (QP) Problems
- Conic Optimisation Problems
- Integer Programming (IP) Problems
- Dynamic Programming (DP) Problems
- Stochastic Programming (SP) Problems

Below is a list of optimisation routine, arranged in order of increasing difficulty for the problem types [86]. Optimisation routine is applied to each problem type to find the solution.

The Simplex Method

The best known (and most successful) methods for solving LPs are interior- point methods and the simplex method.

Performing a pivot of the simplex method is extremely fast on today's computers, even for problems with thousands of variables and hundreds of constraints. This explains the success of the simplex method.

Although the simplex method demonstrates satisfactory performance for the solution of most practical problems, it has the disadvantage that, in the worst case, the amount of computing time (the so-called worst-case complexity) can grow exponentially in the size of the problem.

However, for large problems, the number of iterations also tends to be large. The "large" linear program means a problem with several thousands variables and constraints; say 5,000 constraints and 100,000 variables or more. Such models are not uncommon in financial applications and can often be handled by the simplex method [86].

Interior-Point Methods

After it was discovered in the 1970s that the worst case complexity of the simplex method is exponential (and, therefore, that the simplex method is not a polynomial-time algorithm) there was an effort to identify alternative methods for linear programming with polynomial-time complexity [87]. The more exciting and enduring development was the announcement by Karmarkar in 1984 that an Interior Point Method (IPM) can solve LPs in polynomial time [87].

Interior-point techniques were popular during the 1960s for solving nonlinearly constrained problems. However, their use for linear programming was not even contemplated because of the total dominance of the simplex method [87].

A generalised reduced gradient (GRG) method.

This method and specific implementation have been proven in use over many years as one of the most robust and reliable approaches to solving difficult NLP problems [88].

- General, yet easy to use in put formats and arrange of output Options,
- The ability to solve problems with hundreds of equations

- Dynamic storage allocation, so problems of any size may be attempted by changing only one dimension statement
- A minimum of machine dependent statements, and well documented.

Evolutionary Solver

Evolutionary algorithms (EAs) such as evolution strategies and genetic algorithms have become the method of choice for optimisation problems [89] that are too complex to be solved using deterministic techniques such as linear programming or gradient (Jacobian) methods.

Evolutionary algorithms (EAs) are search methods that take their inspiration from natural selection and survival of the fittest in the biological world. EAs differ from more traditional optimisation techniques in that they involve a search from a "population" of solutions, not from a single point.

The new Evolution solver accepts Solver models defined in exactly the same way as the Simplex and GRG Solvers, but uses genetic algorithms to find its solutions. While the Simplex and GRG solvers are used for linear and smooth nonlinear problems, the Evolutionary Solver can be used for any Excel formulas or functions, even when they are not linear or smooth nonlinear. Spreadsheet functions such as IF and VLOOKUP fall into this category.

EAs have received a lot of attention regarding their potential as optimisation techniques for complex numerical functions. However, they have not produced a significant breakthrough in the area of NLP due to the fact that they have not addressed the issue of constraints in a systematic way [89].

In contrast to linear programming, where the simplex method can handle most instances and reliable implementations are widely available, there is not a single preferred algorithm for solving general nonlinear programmes. Without difficulty, one can find ten or fifteen methods in the literature and the underlying theory of nonlinear programming is still evolving. A systematic comparison between methods is complicated by the fact that a nonlinear method can be very effective for one type of problem and yet fail miserably for another.

Some software packages for solving nonlinear programmes are:

- CONOPT, GRG2, Excel's SOLVER (all three are based on the generalised reduced-gradient algorithm),
- MATLAB optimisation toolbox, SNOPT, NLPQL (sequential quadratic programming),
- MINOS, LANCELOT (Lagrangian approach),

The strategies adopted for this research are discussed in the next sections.

2.7 Research Design

2.7.1 Research methodology adopted in this study

To achieve the aim and objectives of this research, the selection of the research methodology adopted is depicted in Figure 2.3

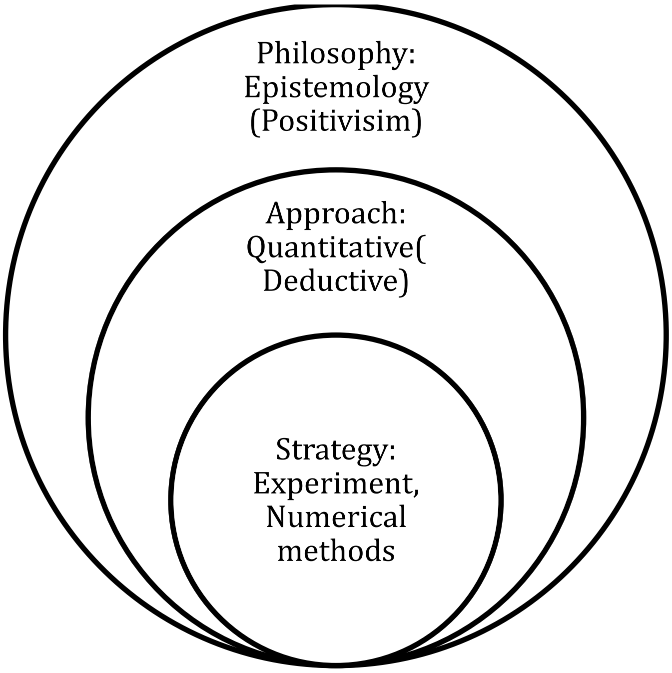


Figure 2.3 Research methodologies applied in this study.

The placement of the research paradigm for this research is epistemological philosophy. Looking at the information above, positivist philosophy seems to be the best fit for this research, as the research focuses on finding the solution on how to optimise the net present value of the project, which falls in positivist paradigm.

This research seems to be using deductive approach as it is considered as scientific research following with theory development and quantitative measurement. Therefore, researcher chose the positivist philosophy, which allow researcher to study on existing literature to gain general knowledge and construct from previous study to avoid the mistake.

Strength: It is a quick method for study development and data collection ensuring with valid data. Secondly, non-return questionnaire is a low risk strategy [67].

Weakness: Deductive approach is constructed with strict methodology and there is no alternative theory. Consequently, the highly construct cause researchers to limit their research design.

Examples of quantitative methods well accepted in the social sciences include survey methods, laboratory experiments, formal methods (i.e. econometrics) and numerical methods such as mathematical modelling, and then submitting the data to scientific techniques for appropriate analysis to test the hypothesis [78, 79].

Although most researchers do either quantitative or qualitative research work, but some researchers have suggested combining one or more research methods in the one study, also called *'triangulation'* [75, 77, 78]. Triangulation refers to the use of more than one approach to investigate a research question(s) in order to enhance confidence in the findings. In this research study, a quantitative research approach seems to be more suitable approach. However, the qualitative approach is used and applied in some part when data collecting and examining case study.

This study uses numerical methods for solving project scheduling problems (PSP) to optimise the NPV. The detail on algorithm and analytical technique will be discussed later on.

2.7.2 Research Process

The initial stage of the research is to develop an understanding of the current practice, in terms of existing scheduling techniques. A desk study of the literature on project scheduling and planning is carried out. The outcomes of the desk study are used to re-define the research problem and influenced the development of a proposed solution as shown in Figure 2.4.

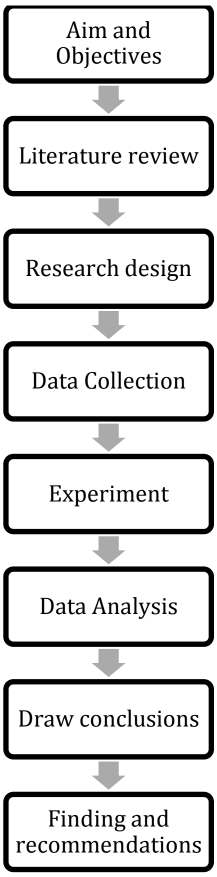


Figure 2.4 Research process flow chart

This study uses numerical method as a research strategy to yield several unique contributions to the research on constrained project scheduling with cash flows. Firstly, the experiment is carried out to test the new heuristic algorithm rule by compare it with the previous rule from literature. Secondly, the rule is measured by employing on large problems, where the data set are obtained from the PTT Company limited in Thailand in which the author has got accessed to during the Industrial placement periods. Lastly, the effectiveness of the heuristic rule is validated through three different projects. An area of strength and weakness in the chosen technique is identified. The more detail on each experiment and algorithm will be discussed further on Chapter 5.

2.7.3 Data collection and analysis

This study uses data for both primary and secondary data. The data are collected from both literature and PTT Public company limited during the author fieldwork in Thailand.

Primary data

On April/May 2011, the author spent between two months duration of the research working as an intern in PTT Company limited, the huge oil& natural gas Company in Thailand. This provides the opportunity to collect primary data from there. This includes internal PTT reports [90] [91] documenting their project data. These reports were available through the company intranet.

Secondary data

Throughout the project, secondary data are identified from a variety of sources. Public domain sources (books and journals) are used to investigate current practices in scheduling project and planning techniques. The literature review can highlight a gap in the existing literature regarding methods of scheduling. The techniques defined in the literature are used to develop a method for scheduling task and improving the NPV of the project.

This research presents a heuristic algorithm with embedded priority rules to optimise the NPV of cash flows for projects. Author proposes a new heuristic rule that improved from the existing one. An experimental design tests the performance of the new heuristic algorithm and compares their performance to that of existing procedures for project scheduling. As mentioned earlier, the data are collected from both literature and PTT Public company limited during the author fieldwork in Thailand. All of the scheduling rules and schemes of the proposed technique will be coded in MATLAB (R2011a). Solver tools of the latest version of Excel spreadsheet will be used to analyse the research data collected and facilitate searching the optimal solution.

2.8 Summary

This chapter describes the overall research process, and in specific the research methodologies and methods adapted to investigate the research problems. The positivist philosophy seems to be the best fit for this research, using deductive approach. The data are collected from both literature and PTT Public company limited during the author fieldwork in Thailand. This study uses numerical methods and carry out experiment in order to identify strength and weakness of the chosen technique in each type of projects.

Chapter 3 Literature Review I: Scheduling techniques

3.1 Introduction

This chapter starts on capital budgeting techniques criticising then follow by a problem characteristic used in this study. Section 3.3 will be an appraising survey of heuristic approaches. Next section will be Schedule Generation Schemes (SGS) that described on how these schemes are employed in priority rule based methods. Section 3.6 is devoted to meta-heuristic algorithms such as simulated annealing, tabu search, and genetic algorithms. Heuristics, which do neither belong to the class of priority, rule based methods nor to meta-heuristic approaches are treated in Section 3.7. The objective of this chapter is to present the review of over all heuristic scheduling techniques.

3.2 Capital Budgeting techniques

This section examines the performance of each capital budgeting technique. NPV and IRR methods are widely used and in some instances. The typical procedure would be use IRR as a screening criterion by testing the project IRR against a minimum hurdle rate [2]. Providing that the project IRR exceeds the hurdle rate, and then the project is considered further, otherwise it is rejected in current form.

The higher the IRR, the more robust the project is, that is the more risk it can withstand before the IRR is eroded down to the level of the cost of capital [92]. If the project IRR does not meet the cost of capital, then the project is unable to repay the cost of financing (assuming it is funded at the normal cost of capital to the company) [93].

Flaig [2] concluded that choosing between projects on the basis of IRR alone risks rejecting higher value projects with a more modest, yet still acceptable rate of return.

For some cash flow patterns and projects the IRR does not exist, and for some others multiple IRRs exist. This happens when there is more than one change in the sign of cash flow [94]. It can even happen that NPV is negative, however the IRR is positive and larger than the cost of capital.

Lurin [95] provided the following decision rules when facing with an investment decision as following;

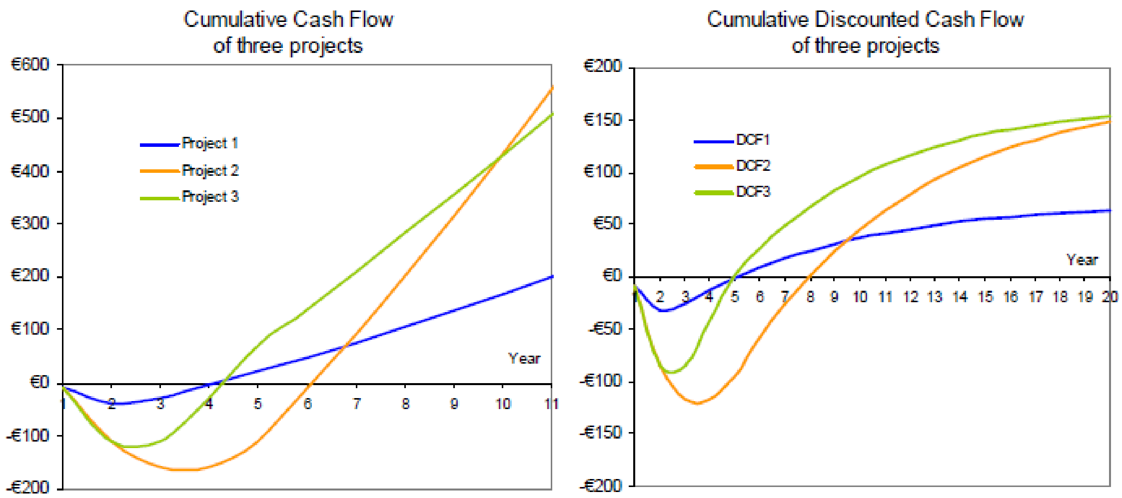


Figure 3.1 Cumulative and Discounted cash flows for three projects from Lurin's book [95]

Table 3.1 NPV and IRR over 20 years for three projects (adapted from [95])

	NPV	IRR
Project 1	€64m	48%
Project 2	€148m	35%
Project 3	€154m	48%

- According to the project profile from Figure 3.1 and Table 3.1, if two projects have the same NPV, choose the one with the higher IRR. It will typically have lower peak cash requirement and/or shorter cash flow payback period. (Project 3 should be preferred to project 2)
- However, very often, a project has a higher NPV but lower IRR (as can be seen on Project 1 and 2 from Table 3.1), which usually means a longer payback period and larger peak-funding requirement for the project with the lower IRR as can be seen on Figure 3.1.
- Also, two projects can have the same IRR but one has a higher NPV (project 1 and 3 from Table 3.1), which can happen with a larger peak funding requirement, the same payback period and a higher positive cash flow for the project with the higher NPV as can be seen on Figure 31.

As these examples show, rather than blindly relying on a single measure such as the IRR, it is better to analyse the cumulative cash flow pattern of the business in particular the peak cash requirement and break-even period. The right way to rank projects is to use the NPV that they generate and their required peak funding, not the IRR [95]. Taking into the consideration of the time value of money, it can be said that the investment analysis tools such as PP and IRR are weak because they do not take into consideration the time value of money [96]. That is, they do not consider the effect of inflation, which can have a significant impact on the results of the analysis. The investment analysis tool that does consider the time value of money is NPV analysis [63]. Some people involved in short projects, which are typically targeted for 6 to 12 months for implementation, might think that the time value of money is not an important issue. However, this would be an erroneous conclusion because, even though the implementation time may be short, the useful project life can be much longer [64]. The financial analysis of projects should be based on the useful life or a long-term view of the project. NPV is seen as a better and superior measure. The PI is also useful where investment capital is a main constraint. It is a measure of capital efficiency, sometimes referred to as the PV ratio.

In conclusion, Flaig [2] compared the aspects of the project highlighted by the techniques discussed so far illustrated in Table 3.2. NPV would probably be the primary method. In a capital constrained environment, the PI would be very important, and if cash flow were a critical issue then payback or IRR would be looked at keenly. This study selects NPV as a chosen technique to evaluate the project performance.

Table 3.2 compares the aspects of the project highlighted by the techniques [2]

Technique	Value	Efficiency	Timing
Payback period	N	Y	Y
PI	N	Y	N
IRR	N	Y	N
NPV	Y	N	N

3.3 The Scheduling Problem

The problem in this research are refer to the precedence constrained scheduling problem as can be described in Thesen's paper [97] as following:

- A set of projects is to be scheduled.
- Each project: consists of a set of activities; has a schedule-dependent duration; once started, should progress at a reasonably consistent rate.
- Within a project, each activity: has a known duration; may not start until certain predecessor activities have finished; should not be interrupted.

From Kolish [98], the problem consists of j = 1, …, n activities with a non-preemptable duration of d_i periods, respectively. The activities are interrelated by constraints: Precedence con strains - as known from traditional CPM-analysis - force an activity not to be started before all its predecessors have been finished.

The objective of this study is to find precedence feasible completion times for all activities such that the NPV of the project is maximised.

Problems containing these elements have been modelled and solved in a wide variety of contexts. Davis [27, 99] and Conway et al. [27] reviewed available approaches up to 1966, Mason and Moodie [100] discussed contributions to 1971. Bennington and McGinnis [101] compared recent algorithms in some details. Davis [4] provided an excellent overview and classification of contributions to the project scheduling field to 1973.

Examples of problem areas containing these elements are readily available, for example, projects such as those referred to by Wiest [102] as "large-one-of-a-kind projects" clearly fall in this category. Included in this group would be any large development project with a clearly defined set of activities as well as with a readily distinguishable beginning and end.

It has been shown by Blazewicz [103] that the problem belongs to the class of NP-hard optimisation problems. Therefore, heuristic solution procedures are indispensable when solving large problem instances as they usually appear in practical cases. Since 1963 when Kelley [104] introduced a schedule generation scheme, a large number of different heuristics algorithms have been suggested in the literature.

From a survey of Kolisch and Padman [105], the great number of optimal approaches are mainly for generating benchmark solutions. They also suggest that the most competitive exact algorithms seem to be the ones of Brucker et al. [106], Demeulemeester and Herroelen [107], Mingozzi et al. [108] and Sprecher [109, 110].

Next section will give an appraising survey of heuristic approaches.

3.4 Priority rule based heuristics

A priority rule based scheduling approach consists of two components, a priority rule to determine the list with the rankings of activities and a schedule generation scheme (SGS) to construct a feasible project schedule based on the constructed activity list [111]. In Figure 3.2, the approach is illustrated graphically and shows that the project data is used to construct a list of activities using a priority rule, which is then transformed by a SGS into a feasible project baseline schedule

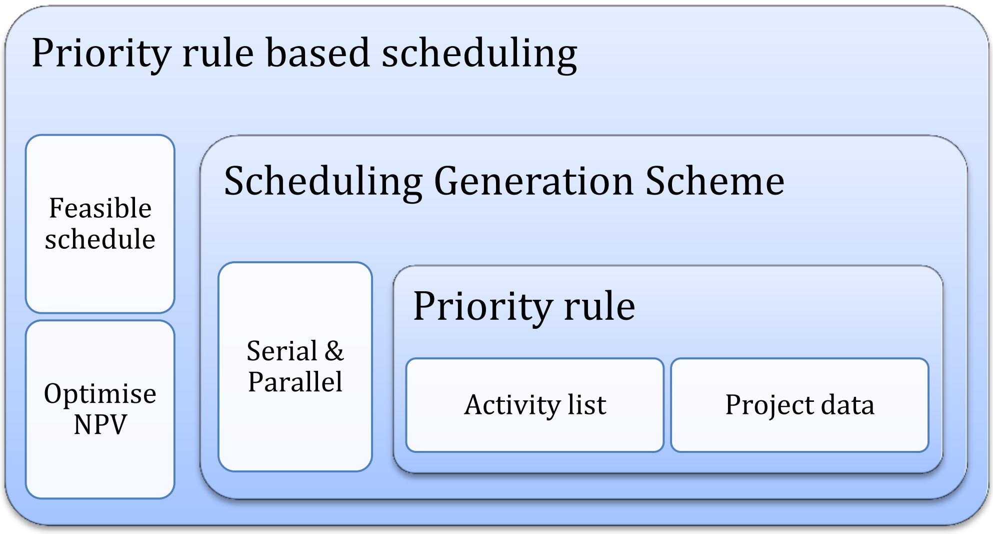


Figure 3.2 The priority rule based scheduling approach to construct a feasible project schedule.

3.4.1 Priority Rules

A priority rule contains information to construct a list of activities that ranks all project activities in a certain order to determine the priorities in which the activities are assigned to the project schedule. Such a list is constructed based on the project data in order to assign priorities to activities [112], as follows:

- Activity information: information about time or cost estimates of the activities determines the activity priorities.
- Network information: information on the project network logic determines the activity priorities.
- Scheduling information: information obtained from simple critical path scheduling tools determines the activity priorities.

3.4.2 Schedule generation scheme (SGS)

Kelley [104] introduced a SGS which determines the way in which a feasible schedule is constructed by assigning start times to the project activities. At the start of the heuristic scheduling process, the partial schedule is empty and all activities are available to be scheduled. Afterwards, activities are selected according to their priorities and are put in the schedule following the rules of the SGS. Basically, two well-known SGS are available, as follows:

- Serial schedule generation scheme (SSGS): selects the activities one by one from the list and schedules it as-soon-as-possible in the schedule.
- Parallel schedule generation scheme (PSGS): selects at each predefined time period the activities available to be scheduled and schedules them in the list as long as enough resources are available.

The Serial Schedule Generation Scheme (SSGS)

The serial method proposed by Kelley [104]. An activity is selected one at a time and as soon as possible within the precedence and resource constraints. To that purpose, the scheme scans the priority list and selects at each stage the next activity from the priority list in order to schedule it at its first possible starting time without violating both the precedence and resource constraints [113].

The Parallel Schedule Generation Scheme (PSGS)

A parallel schedule generation scheme proposed by Kelley [104] as well, iterates over the time horizon of the project instead of iterating over the priority list and adds activities that are eligible to be scheduled. More precisely, the scheme starts at time point t = 0 and schedules activities before the time pointer is increased. It selects at each decision point t the eligible activities and assigns a scheduling sequence of these eligible activities according to the priority list. At each decision point, the eligible activities are scheduled with a starting time equal to the decision point (on the condition that there is no resource conflict). Activities that cannot be scheduled due to a resource conflict are skipped and become eligible to schedule at the next decision point t’ > t, which equals the earliest completion time of all activities active at the current decision point t.

Priority rule based heuristics combine priority rules and schedule generation schemes in order to construct a specific algorithm. If the heuristic generates a single schedule, it is called a single pass method, if it generates more than one schedule, is referred to as multi pass method.

Single Pass Methods

The oldest heuristics are single pass methods, which employ one SGS and one priority rule in order to obtain one feasible schedule. Recently, more elaborate priority rules have been proposed.

Multi Pass Methods

There are many possibilities to combine SGS and priority rules to a multi pass method. The most common ones are multi priority rule methods, forward-backward scheduling methods, and sampling methods.

Forward-backward scheduling methods

These techniques employ an SGS in order to iteratively schedule the project by alternating between forward and backward scheduling [114]. The priority values are usually obtained from the start or completion times of the lastly generated schedule. Forward-backward scheduling methods have been proposed by, i.e. Li and Willis [115] as well as Ozdamar and Ulusoy [116, 117].

Sampling methods

These methods make generally use of one SGS and one priority rule. Different schedules are obtained by biasing the selection of the priority rule through a random device [1] [118]. Instead of a priority value, a selection probability value is computed. Dependent on how the probabilities are computed, one can distinguish random sampling, biased random sampling, and regret based biased random sampling [107]. Biased random sampling methods have been applied by Alvarez-Valdes and Tamarit [119] and Cooper [120].

3.5 Priority rules developed in project scheduling

Below, a summary of the most commonly used priority is given for each of the first four classes [104]. It should be noted that this is certainly an incomplete list of priority rules since one can think of many other priority rules or extensions or combinations of these rules.

3.5.1 Activity information

The construction of an activity list is based on a priority rule taking the characteristics of the project activities into account, such as the duration of each activity. Example priority rules are:

- Shortest Processing Time (SPT): Put the activities in an increasing order of their durations in the list [121].
- Longest Processing Time (LPT): Put the activities in a decreasing order of their durations in the list [122].

3.5.2 Network information

The construction of an activity list is based on a priority rule taking the logic of the network structure into account, i.e. the set of activities and the precedence relations between them. Example priority rules are:

- Most Immediate Successors (MIS): Put the activities with the most direct successors first in the activity list [123].
- Most Total Successors (MTS): Put the activities with the most direct and indirect successors first in the activity list [124].
- Least Non-Related Jobs (LNRJ): A job (or activity) is not related to another job if there is no precedence related path between the two activities in the project network [125].

- Greatest Rank Positional Weight (GRPW): The GRPW is calculated as the sum of the duration of the activity and the durations of its immediate successors [126] [127].

3.5.3 Scheduling information

Priority rules are used to construct feasible project schedules with resource constraints. However, simple scheduling techniques that ignore these resource constraints, such as the critical path method, can also be used to define new priority rules [128]. Example priority rules are:

- Earliest Start Time (EST): Put the activities in an increasing order of their earliest start in the list [129].
- Earliest Finish Time (EFT): Put the activities in an increasing order of their earliest finish in the list [130].
- Latest Start Time (LST): Put the activities in an increasing order of their latest start in the list [131].
- Latest Finish Time (LFT): Put the activities in an increasing order of their latest finish in the list [61, 132].
- Minimum Slack (MINSLK): Put the activities in an increasing order of their slack value in the list [133].

3.6 Meta-heuristic approaches

Several meta-heuristic strategies (sometime are called optimisation-based heuristics) have been developed to solve hard optimisation problems [134]. The following summary briefly describes those general approaches that have been used to solve the RCPSP.

3.6.1 Simulated Annealing

Simulated Annealing (SA), introduced by Kirkpatrick et al. [135], originates from the physical annealing process in which a melted solid is cooled down to a low-energy state. Starting with some initial solution, a so-called neighbour solution is generated by slightly perturbing the current one. If this new solution is better than the current one, it is accepted, and the search proceeds from this new solution. Otherwise, if it is worse, the new solution is only accepted with a probability that depends on the magnitude of the deterioration as well as on a parameter called temperature. As the algorithm proceeds, this temperature is reduced in order to lower the probability to accept worse neighbours.

Clearly, SA can be viewed as an extension of a simple greedy procedure [136], sometimes called First Fit Strategy (FFS), which immediately accepts a better neighbour solution but rejects any deterioration.

3.6.2 Tabu Search

Tabu Search (TS), developed by Glover [137, 138], is essentially a steepest descent/mildest ascent method. That is, it evaluates all solutions of the neighbourhood and chooses the best one, from which it proceeds further.

This concept, however, bears the possibility of cycling, that is, one may always move back to the same local optimum one has just left. In order to avoid this problem [139], a tabu list is set up as a form of memory for the search process. Usually, the tabu list is used to forbid those neighbourhood moves that might cancel the effect of recently performed moves and might thus lead back to a recently visited solution. Typically, such a tabu status is overrun if the corresponding neighbourhood move would lead to a new overall best solution (aspiration criterion).

It is obvious that TS extended the simple steepest descent search, often called Best Fit Strategy (BFS), which scans the neighbourhood and then accepts the best neighbour solution, until none of the neighbours improves the current objective function value [140].

Icmeli and Erenguc [132] applied a tabu search procedure to a starting feasible solution generated using a simple single-pass algorithm. The initial solution was improved over several iterations by moving each activity one time unit early or late from its current completion time, with the restriction that the resulting completion time should not violate earliest and latest completion times for the activity. They also investigated the use of long-term memory within tabu search to further improve the results.

Computational results on 50 problems from the Patterson set indicated that these procedures were both efficient and close to optimal.

3.6.3 Genetic Algorithms

Genetic Algorithms (GA), inspired by the process of biological evolution, have been introduced by Holland [141]. In contrast to the local search strategies above, a GA simultaneously considers a set or population of solutions instead of only one [142]. Having generated an initial population, new solutions are produced by mating two existing ones (crossover) and/or by altering an existing one (mutation). After producing new solutions, the fittest solutions survive" and make up the next generation while the others are deleted [143]. The fitness value measures the quality of a solution, usually based on the objective function value of the optimisation problem to be solved [144].

3.7 Other heuristics

3.7.1 Truncated Branch and Bound Methods

Pollack-Johnson [145] used a so-called depth-first, jump tracking branch and bound search of a partial solution tree. The algorithm is essentially a parallel scheduling heuristic [146]. Instead of scheduling the activity with the highest priority value it branches on certain occasions such that one branch has the activity with the highest priority value and the other branch has the activity with the second highest priority value, which is scheduled next. Note, due to use of the PSGS optimal solution might be excluded from the search space [147].

Sprecher [110] employed his depth-first search branch and bound procedure as a heuristic by imposing a time limit. The enumeration process was guided by the so-called precedence tree, which essentially branches on the activities in the decision set of the SSGS. Via backtracking, all precedence feasible activity lists are (implicitly) enumerated. In order to obtain good solutions early in the search process (and thus within the time limit), priority rules were applied to select the most promising activity from the decision set for branching first [148].

3.7.2 Disjunctive Arc Based Methods

The basic idea of the disjunctive-arc-based approaches is to extend the precedence relations (the set of conjunctive arcs) by adding additional arcs (the disjunctive arcs) such that the minimal forbidden sets, i.e. sets of technologically independent activities which cannot be scheduled simultaneously due to resource constraints, are destroyed and thus the earliest finish schedule is feasible with respect to (precedence and) resource constraints [149].

Shaffer et al. [37] restricted the scope, within their "resource scheduling method", to those forbidden sets for which all activities in the earliest finish schedule are processed at the same time. The disjunctive arc, which produces the smallest increase in the earliest finish time of the unique sink, was introduced and the earliest finish schedule was recalculated. The algorithm terminates as soon as a –precedence- and resource-feasible earliest finish schedule is found.

Alvarez-Valdes and Tamarit [119] proposed four different ways of destroying the minimal forbidden sets. The best results were achieved by applying the following strategy: Beginning with the minimal forbidden sets of lowest cardinality, one set is arbitrarily chosen and destroyed by adding the disjunctive arc for which the earliest finish time of the unique dummy sink is minimal.

Bell and Han [150] presented a two-phase algorithm for this problem. The first phase was very similar to the approach of Shaffer et al. However, phase 2 tried to improve the feasible solution obtained by phase one as follows: after removing redundant arcs, each disjunctive arc that was part of the critical path(s) was temporarily cancelled and the phase 1 procedure was applied again.

3.7.3 Further Approaches

Integer programming based heuristics have been used by Oguz and Bala [151]. The method employs the integer programming formulation originally proposed by Pritsker et al. [152]. The planning horizon is divided in T periods of equal length and the processing times p_j have to be given as discrete multiples of one period. The binary decision variable is $x_{j,t} = 1$ if activity j is finished at the end of period t [153].

Mausser and Lawrence [139] used block structures to improve the makespan of projects. They started by generating a feasible solution with a parallel scheduling scheme. Following this, they identified blocks, which represent contiguous time spans that completely contain all activities processed within it. Each such block can be considered independent of the other blocks. The method essentially rescheduled individual blocks in order to shorten the overall project length [154].

Zhu and Padman [155] also applied distributed computing concepts to the RCPSP through the use of an Asynchronous Team (A-Team) approach. An A-team is a software organisation that facilitates cooperation amongst multiple heuristic algorithms so that together they produced better solutions than if they were acting alone. They embedded several simple heuristics for solving the RCPSP within the iterative, parallel structure of A-Team, which provide a natural framework for distributed problem solving. Preliminary results on small randomly generated project networks indicated that the combination of multiple, simple heuristics outperform many single-pass, complex optimisation-based heuristics proposed in the literature.

3.8 Summary

The problem of scheduling activities under constraints is a relatively common problem that has received considerable attention in the literature. Unfortunately, at this time, optimal solutions can be found only for unrealistically small problems of marginal practical value [97]. The NP-hard nature of the problem makes it difficult to reach an exact solution for realistic-sized projects. Hence, in practice, the use of simple heuristics is necessary. They are based on a process of decision making according to a set of priority rules that are based on activity characteristics [50]. Next chapter will be focusing on the development of scheduling techniques with max NPV.

Chapter 4 Literature Review II: Maximising NPV

4.1 Introduction

This chapter offers a guided tour through the important recent developments in the expanding field of research on project scheduling problems. Proper attention is given to NPV maximisation models for the project scheduling problem with known cash flows, optimal and suboptimal scheduling procedures with various types of resource constraints, and the problem of determining both the timing and amount of payments.

The objective of this chapter is to critically review the various contributions, which try to capture the monetary and financial objectives of the project scheduling problem in the form of the maximisation of the NPV. Data described in this chapter have been reported and published before in author's own paper [156] [157].

4.2 Project Scheduling with max NPV on Literature

In 1970, A.H. Russell [158] was the first to introduce the objective of 85aximizing the NPV of cash flows in a network. Russell deal with the unconstrained problem where both positive and negative payments occur as events in the project are completed.

A project consists of a set of activities. The performance of each activity involves a series of cash flow payments and receipts throughout the activity duration. A terminal value of each activity upon completion can be calculated by discounting the associated cash flows to the end of the activity.

Equation (4.1) NPV objective function [158]

$$maximise\ NPV = \sum_{i=1}^{n} CF_i \exp(-\alpha T_i) \qquad (4.1)$$

where

exp(-α)	=	*β*, the discount factor.
CF_i	=	Cash flow for activity i

For uniformity of expression, the criterion Eq. (4.1) is sometimes rewritten as:

Equation (4.2) NPV formula (rewritten as discount factor)[158]

$$maximise\ NPV = \sum_{i=1}^{n} CF_i \beta^{T_i} \qquad (4.2)$$

Russell transformed the nonlinear objective function into a linear one by approximation using the first term of the associated Taylor series expansion. He does not report computational results with his procedure apart from two small example problems, although reference is made to a computer programme being developed to solve this problem. Because of the development of fast and efficient network computer codes since the publication of this paper, there do not seem to be any theoretical obstacles to implementing his approach. His research showed that the cost-critical path is quite different from the time-critical path when monetary objectives are considered.

Grinold [159] transformed the unconstrained problem formulated by A.H. Russell into an equivalent linear programming problem in 1972. This problem was exploited by the solution procedure that determines the optimal solution by exploring the set of feasible trees on the project network such that all activities have zero slack. This procedure was also used to illustrate, with an example, the trade-off between NPV and project duration He does not provide extensive computational results for his procedure.

In 1990, Elmaghraby and Herroelen [160] critiqued both Russell's and Grinold's formulations to develop a simplified algorithm that gives the optimal schedule for the project scheduling problem with NPV objective. They showed that, in general, it is optimal to schedule events with associated positive cash flows as early as possible, and events with net negative cash flows as late as possible subject to restrictions imposed by network structure. They also illustrated that net cash flows are dependent on the time of realisation of cash flow nodes and in the absence of a project deadline, if the NPV is less than zero, the project will be delayed indefinitely.

Demeulemeester et al. [161] proposed a new optimal algorithm in 1996 that performs a recursive search on partial tree structures that utilised the concept of scheduling activities early if they bring in payments and delaying those activities that incur expenses. Computational tests reported encouraging results in comparison to the Grinold procedure.

Doersch and Patterson [162] were the first to study in the context of the resource-constrained max-npv problem in 1977. They introduced a binary integer programming approach to the NPV project scheduling problem. This model included a constraint on capital for expenditure on activities in the project such that the available capital increased as progress payments were made. The objective function also included the cash flows associated with the completion of activities and any penalties incurred for late completion.

The terms in the objective function represent cash outflows, cash inflows and capital costs, respectively, where each component is discounted back to the beginning of the project:

Equation (4.3) Max NPV model from Doersch and Patterson [162]

$$Maximise \sum_{k=1}^{m} CF_{i(k)} \exp\left(-\propto T_{i(k)}\right) + CF_{j(k)} \exp\left(-\propto T_{j(k)}\right)$$

$$+[I_k \exp(-\propto d_k) - I_k]\exp(-\propto T_{i(k)}) \qquad (4.3)$$

where

CF_0 = total capital available at beginning of the project in period 0.

I_k = capital investment required by activity k, k = 1, 2, …, m.

$CF_{i(k)}$ = cash outflow at the beginning of activity k at node i, where each activity is defined by nodes i and j.

$CF_{j(k)}$ = cash inflow received upon completion of activity k at node j.

d_k = duration of activity k, where k may not be preempted.

$T_{i(k)}$ = time at which node i of activity k is scheduled to occur.

$Z_{a(t)}$ = the set of activities a that are scheduled to be active in period t.

$Z_{p(t)}$ = the set of activities p, completed prior to period t.

$\propto$ = Opportunity cost of capital.

Equation (4.4) The activity precedence constraints of NPV [162]

$$T_{j(k)} - T_{i(k)} \geq d_k, \ k = 1,2, \ldots, m. \qquad (4.4)$$

As the project is enacted, the net capital balance reflects positive and negative cash flows associated with activities and nodes completed in previous periods. As in Doersch and Patterson [162] assume that capital is a renewable resource, where the initial capital availability CF_0 is augmented or reduced by the cash flows that occur throughout the project. Thus, the capital constraints for each period of the project are

Equation (4.5) The capital constraints for each period of the project [162]

$$\sum_{k \in Z_{a(t)}} I_k \leq CF_0 + \sum_{k \in Z_{p(t)}} (F_{j(k)} + F_{i(k)}) \qquad (4.5)$$

The model was solved to optimality for projects involving 15–25 activities. The results indicated that at high cost of capital or long project duration, it is important to evaluate bonus/penalty and capital constraints while scheduling activities. However, detailed computational results are not provided.

In 1987, Smith-Daniels and Smith-Daniels [163] extended the Doersch and Patterson [162] zero-one formulation to accommodate material management costs. The NPV of the project was maximised. They concluded that not only do ordering and holding cost force activities with common requirements to start at the same time or close to each other, the additional constraints also resulted in lowering overall project cost even though they may cause activities, and hence the project, to be delayed.

In 1986, Tavares [164] proposed a new dynamic programming formulation and solution method, where the optimality conditions were derived using calculus of variations for a set of interconnected projects. The objective function to be maximised included a net of the discounted sum of the benefits generated along the programme, the discounted sum of the cost of project expenditures, and a term to penalise the variation in expenses over time. This programme was applied successfully to a large railway construction project in Portugal.

Patterson et al. [147] presented a zero–one programming model and a backtracking algorithm in 1990 to maximise the NPV of the constrained project scheduling problem in 1990. It is unique in that it can also be used to minimise project duration. The solution methodology utilised the fact that the minimum duration problem is easier to solve than the max NPV problem and used it as a heuristic to generate starting solutions on which right-shifting of cash flows was applied to improve NPV. Problems, ranging from 10 to 500 activities, were tested on both objectives using MINSLK and random rules, with optimal solutions found only for the smaller problems. The MINSLK rule generated higher NPV than the random rule.

Baroum and Patterson [165] proposed a new heuristic scheduling rule to solve the project scheduling problem with cash flow in 1996.

Talbot and Patterson [166] ordered the activities in an activity list such that precedence relations are taken into account. Then, they derived time windows for all activities by forward recursion from t = 0 and backward recursion from an upper bound of the makespan. Starting with the first activity on the list, the enumeration process tried to schedule the next activity on the list at the earliest precedence and resource-feasible interval within the activity-specific time window. If this is not possible, backtracking occurs to the last activity, which is then scheduled one period later. The basic enumeration was enhanced by network cuts, which allowed pruning a part of the enumeration tree.

In 1992, Yang et al. [167] developed an integer programming approach for the NPV objective which was based on the solution procedure of Talbot and Patterson [166]. The latter approach has been designed for the makespan objective. The proposed procedure, however, solves only problems with a small number of activities. For solving larger problems with many activities, the optimal procedure requires excessive computation time and heuristic rules are the only currently available viable solution procedures.

Icmeli and Erenguc [126] also developed a branch-and-bound algorithm in 1996 for the RCPSP with cash flows which used the minimal delaying alternatives concept for branching. This concept together with the rule that determines the node to branch from are used in bounding the size of the tree. The algorithm was tested on 50 test problems from the Patterson set with the number of activities ranging from 7 to 51, and 40 problems with 32 activities generated using ProGen, and with up to 3 resource types and was shown to be efficient in comparison to results in the literature.

Kolisch and Drexl [168] proposed a special multi-pass approach, called adaptive search procedure in 1996. The method makes use of the serial and parallel schedule generation scheme employing a deterministic as well as a sampling method. Based on an analysis of the problem at hand and the number of iterations already performed, the procedure decides on the specific method to apply. The use of bounds lowers the computational effort.

In 1994, Özdamar and Ulusoy [117] embedded 'local constraint- based analysis' into a single-pass parallel scheduling scheme in order to decide which activities have to be scheduled and which activities have to be delayed at any given time, via feasibility checks and the so-called 'essential conditions'.

Russell [169] provided one of the first comparisons of heuristics for scheduling projects with resource constraints where the objective is to maximise project NPV. He introduced priority rules for selecting activities for resource assignment based upon information derived from the optimal solution to the unconstrained problem. He used both insight from the relaxed resource-unconstrained NPV problem and methods designed for the minimum duration problem to develop six heuristics. They were tested on 80 problems ranging from small-scale problems with 30 activities to large-scale problems with 1461 activities. One of the heuristics, based on random selection of activities for scheduling (RAND-50), was used as a benchmark to select the best out of 50 randomly generated solutions. It was observed that no one specific heuristic performed best in all situations. For the small-scale problems, the heuristics had similar performance and were within 5 -10% of the optimal solution. As the project size increased, the level of resource-constrained determined the efficient heuristics. The minimum slack rule with the lowest activity number as tie breaker (MINSLK/LAN), a good rule for the minimum duration problem, was found to perform best for large projects when the resource constraints were not tight. In contrast, when resources are tight, rules based on the relaxation of the RCPSP provided better performance, additionally reinforcing the fact that max NPV problem requires new approaches compared to the minimum duration problem. In conclusion, two observations were made. When resources are loosely constrained, the minimum activity slack scheduling rule performs best. When resources are tightly constrained, the three heuristic rules that use information obtained from the optimal unconstrained solution perform best. The differences in performance among the rules were, however, not statistically significant.

Padman et al. [170] developed heuristic procedures in 1997 to schedule projects with multiple constrained resources. They showed that a heuristic procedure with embedded priority rules that uses information from the repeated solution of the relaxed optimisation model to increased project NPV. The heuristic procedure and nine different embedded priority rules were tested in a variety of project environments that considered different network structures, levels of resource-constrainedness, and cash flow parameters, called the PSD data set. Extensive testing on the PSD data set showed that the new heuristic procedures dominate heuristics using information from the Critical Path Method (CPM) and in most cases outperform heuristics from previous research. The best performing heuristic rules classified activities into priority and secondary queues according to whether they led to immediate progress payments, thus front loading the project schedule.

Padman and Smith-Daniels [171] extended previous work using the relaxed optimisation model to evaluate trade-offs between early and tardy penalties in the scheduling of activities. Another eight heuristic rules based on the computed early and tardy penalties are proposed. They embedded eight heuristics in the greedy procedure discussed to test whether releasing activities to the schedule queue as soon as their predecessor activities were completed could result in improved project NPV. Computational results showed that the newly proposed early schedule heuristic rules perform better than the previously examined target-time heuristic rules. Extensive testing on the PSD data indicated the success of this approach.

Smith-Daniels and Aquilano [38] considered the resource constrained max-npv problem. They compared the duration and NPV of a late-start critical path schedule to that of an early-start critical path schedule. It was assumed that cash outflows occurred at the beginning of the period and a single project payment was received on completion of the project. Their assumptions were tested using the 110 Patterson problems. An improved average NPV and lower average duration can be found for late-start schedules than early-start schedules. They concluded that a heuristically determined right shifted schedule yields a higher NPV and lower average duration than schedules derived with heuristics that schedule each activity as early as possible.

Ulusoy and Özdamar [116] presented an iterative scheduling algorithm with the objective of improving both the project duration and NPV in 1996. The consecutive forward/backward scheduling passes made by the iterative algorithm result in a smoother resource profile, which, along with right-shifting of activities, improves both the project duration and NPV. In the cash flow model assumed here, activity expenditures occur at their starting times and payment is made on completion of the project. The algorithm was tested on two sets of problems from the literature. The results demonstrated that under the assumed cash flow model, the iterative scheduling algorithm improved both criteria.

Zhu and Padman [172] adapted multi-heuristic combination for solving project scheduling problems in 1997. They called up randomly six simple rules that capture different aspects of the scheduling problem, such as resource-constrainedness and network topology, to schedule activities. The underlying premise was that over a number of iterations, the rules would exploit the changing conditions in the project environment. Extensive experimentation conducted using the Patterson and PSD data sets reveal the superior performance of the combination method in comparison with the individual participants. Learning strategies, the most natural extension, were not incorporated in this study.

The following year Zhu and Padman [173] also reported on the design, implementation, and experimentation of a local search enhancement strategy for schedule improvement using tabu search. The procedure, using cash flow-based move generation strategies, helped to overcome the problems associated with getting trapped in local optimal and was equally useful as a repair heuristic. Several parameters within tabu search, such as novel candidate generation strategies, were examined and their impact on solution methods and project NPV were evaluated. Unlike previous heuristics, the meta-heuristic approach dominated in over 85% of the PSD problems, a significant improvement over heuristics in the literature. The results illustrated that problem-independent, meta-heuristic approaches were better able to exploit the complex interactions of the many critical parameters of the RCPSP in comparison to the single-pass, parameter-based, problem-dependent heuristics that were commonly used.

Smith-Daniels et al. [174] argued that the capital constrained project scheduling problem presented a unique managerial challenge as compared to the RCPSP since, in large projects, it is frequently the case that a capital constraint limits the value of work that may be put in progress at any time. In contrast to the RCPSP, additional quantities of capital, the constrained resource, become available for use as progress payments are received for completed work. Since the objective is to maximise project net present value, it is important for the project schedule to arrive at a balance between early receipt of progress payments, which improve NPV and increase the capital balance available, and delay of particular large expenditures. Heuristic methods, using information from the solution to the unconstrained NPV problem, were tested on large project networks, presenting the first results on this practical problem.

Erenguc et al. [175] pointed out that in previous formulations of the RCPSP with cash flows, the activity durations were assumed to be fixed and reductions in the activity durations were not allowed. They presented the time-cost trade-off problem where the durations can be reduced from their normal requirements by allocating more resources, assumed to be unlimited, with associated crashing cost that were included in the NPV objective function. They developed a generalised Benders decomposition procedure for obtaining an optimal solution. This procedure was tested on 56 problems with reasonable computational effort.

Boctor [122] employed a modified parallel scheduling scheme, where an activity was in the decision set if it was at least resource feasible in one mode. Activities were chosen with the MINSLK-rule, and modes were chosen on account of the minimum duration. A multi-pass variant used five ordered pairs of activity- and mode-priority rules.

Instead of choosing one activity from the decision set, Boctor, was chosen the set of nondominated schedulable activities by calculating a lower bound of the prolongation of the resource-unconstrained makespan.

Özdamar and Ulusoy [176] broadened their ‘local constraint-based analysis’-approach to solve the multi-mode RCPSP. They reported results which were consistently better than the single-pass priority rule-based approaches and a multi-pass approach, respectively.

Kolisch and Drexl [177] suggested a local search procedure which especially takes into account scarce nonrenewable resource. The method employs a look-ahead strategy to obtain an initial feasible mode-assignment, i.e. an assignment of each activity to one of its modes, followed by a basic local search performed on the mode-assignments. Every feasible mode-assignment was evaluated by running the adaptive search algorithm of Kolisch and Drexl.

Bey, Doersch and Patterson[11] argued that since the decision to organise on a project basis often is an indication that a firm is committing substantial portions of its financial resources to relatively few projects, the effective timing of cash receipts and outlays can have a significant impact on the ultimate profitability of the endeavor. And even in the case of a relatively small contractor, opportunities do exist for increasing profitability through the judicious scheduling of progress payments. This problem, which is equally relevant for contractors and clients alike, is called the Payment Scheduling Problem.

As pointed out by Elmaghraby [118], the use of network models as aids in the preparation of project bids has received little research attention, even though cost estimation and bidding have been popular topics with practitioners for a fairly long time. In his paper, Elmaghraby suggested a method of arriving at the project cost based on the expenses associated with each activity in the project and the activity schedule. Each milestone event in the project was allocated some of the cost of all activities that precede the event, and the activity schedule was used to adjust for the time value of money.

Dayanand and Padman [7] further discussed the problem of determining the amount and timing of payments from a contractor’s perspective. Optimal and heuristic payment schedules to an integer programming model were shown to be affected by a number of factors such as project deadlines, the number of payments, profit margins, cost of capital, pattern of expenses and the structure of the network. In particular, when progress payments were based on expenses incurred by the contractor, the percentage of expenses recovered with each payment and the number of payments have a significant impact on payment schedules.

Dayanand and Padman [178] also proposed a multistage heuristic to determine a set of payments using simulated annealing in the first stage. In the second stage, activities were rescheduled to improve project NPV. The performance of this general purpose heuristic was compared with other problem-dependent heuristics with significant improvement in schedules and NPV.

Tormos and Lova [179] applied a forward backward improvement technique (also called justification) to improve schedules constructed by heuristics. This simple procedure the activities were shifted to the right within the schedule and then to the left produced excellent results and can be combined with almost any other approach. It can be expected that forward–backward improvement (FBI) will become an important component in future heuristics for the RCPSP.

Aquilano and Smith [180] listed a set of algorithms for finding project schedules subject to activity durations, precedence constraints, and material lead times and inventories. The technique, which they called CPM-MRP, uses a Materials Requirements Planning-like bill of materials and schedule format to list the project network and project schedule. Requirements for non-storable resources such as labour and equipment were listed, but the project schedule was not found subject to constraints on the availability of these resources.

Smith-Daniels and Aquilano [181] developed a system which provides a complete integrated project scheduling device. The system is based on treating activities, resources and material as entities in a Material Requirement Planning (MRP) type logic and exploding the 'project Bill Of Material' (BOM).

Lee and Khumawala [15] listed a Material Requirements Planning-type system that is designed to schedule large projects such as NASA's space shuttle. They described a technique that utilised a project bill of materials to schedule multiple projects (space shuttle flights) in a serial fashion subject to constraints on non-storable resource availability. Material lead times and inventories were not in the constraint set for this model.

4.3 Summary

Past researches [158] [159] [162] have developed many different deterministic, single-pass heuristic decision rules for maximising project NPV. Due to the limitation of these single-pass rules is that they only generate a single solution or schedule for a problem, many researches focus has been on meta-heuristics. Genetic algorithms and tabu search have been the most popular strategies for the last ten years. Moreover, the first application of ant systems to the RCPSP as well as various non-standard local-search and population-based schemes have been proposed [182]. The activity list has been the most widely used representation. It has usually been employed in its classical form, while a few researchers have extended it.

From the work done by [15, 180, 181], the result is a PERT/MRP network that can be processed simultaneously with other projects to develop a feasible production schedule for the plan that does not violate any capacity constraints, material availability, or overall lead times. However this scheduling technique is useful in cases where complex products of the same family are manufactured simultaneously in small quantities. The technique is not applicable in the case of a project where the bill of material is not known early in the life of the project and lead-time for some special-purpose components is long.

Among the papers reviewed, Russell [169] have briefly evaluated the performance of heuristic scheduling rules. Two observations were made that MINSLK scheduling rule performs best when resources are loosely constrained and no one specific heuristic performed best in all situations. Padman and Smith-Daniels [171] and Padman et al. [170] also compared heuristics rules which are similar to the single-pass heuristics for maximising the NPV of cash flows in a resource-constrained network. They have devised a new scheme for imbedding single pass heuristics in a forward-pass, network flow procedure that dynamically updates information from the A.H. Russell [169] formulation.

Considering the development during the last years, priority rule-based methods have become attracted more attention than other new approaches [183] [184]. Zhu and Padman [155] revealed that the combination of simple heuristics mostly outperform complex optimisation-based heuristics proposed. The general observation is that the new propose techniques contain more components than earlier procedures but fail to apply with the more complexity of the project. Although optimisation-based approaches usually produce the optimal results than priority rule-based heuristics, they fail to solve the relatively medium-size and more complicated problems usually encountered in practice [185]. In addition, their application may result in a considerably higher computational effort [186]. Priority rule-based heuristics are in wide and general use due to yielding acceptable results with a reasonable computational effort and can be applies in realistic problems. It is a very good idea to employ more effective scheduling schemes within such procedures.

This research is proposed the new rule that combines the simplicity of the priority rule scheduling but at the same time adding more components. Many methods consider both scheduling directions instead of only forward scheduling, more than one type of local search operator, or even more than one type strategy [187, 188]. While recombining merely existing ideas occasionally seems to be less creative than developing new ideas, some of the integration efforts have put well-known techniques into a new and promising context, and the results have often been encouraging.

Chapter 5: [m-CCF] Heuristic Algorithm

5.1 Introduction

This chapter provides an overview and procedure of proposed technique used in this study. m-CCF scheduling is a heuristic scheduling technique that makes use of two components to construct a feasible project schedule, a priority rule and a schedule generation. In this chapter, the use of two alternative schedule generation schemes along with experimental design are described and illustrated.

5.2 Problem description and Assumptions

The problem consists of j = 1, 2, 3, …, J activities with a duration of d periods, respectively. The activities are interrelated by constraints: Precedence con strains - as known from traditional CPM-analysis – force an activity not to be started before all its predecessors have been finished. The capital constraint is usually imposed on a project to limit the amount of capital that may be expended per period. Since capital is limited, activities might not be scheduled at the earliest (precedence feasible) start time but later.

The objective of this study is to schedule the activities such that precedence and capital constraints are obeyed and the NPV of the project is maximised.

In order to model the problem: Let CF_j denotes cash flow of activity j; t_j denotes time at which activity j is scheduled to occur; P_j denotes the set of immediate predecessors of activity j; I denotes the capital available and α denotes the discount rate. Now, a conceptual model can be formulated as follows;

Equation (5.1) NPV objective function [32]

$$maximise\ NPV = \sum_{j=1}^{J} CF_j \exp(-\alpha t_j) \qquad (5.1)$$

subject to

Equation (5.2) Precedence constraints function [98]

$$FT_i \leq FT_j - d_j\ ,\ \ j = 2, \dots, J, \quad i \in P_j \qquad (5.2)$$

Equation (5.3) Precedence constraints function [98]

$$FT_j \geq 0\ ,\ \ j = 1, \dots, J \qquad (5.3)$$

Equation (5.4) Capital constraints function [98]

$$\sum_{j \in A_t} CF_j \leq I \qquad (5.4)$$

The variable FT_j denotes the finish time of activity j, j = 1, …., J and A_t denotes the set of activities being in progress in period t as

Equation (5.5) Active set function [98]

$$A_t = [\, j \mid j = 1, \dots, J, FT_j - d_j + 1 \leq t \leq FT_j \,] \qquad (5.5)$$

The objective function (5.1) maximises the NPV of the project. Constraints (5.2) and (5.3) take into consideration the precedence relations between each pair of activities (i, j), where i immediately precedes j. Finally; constraint set (5.4) limits the total capital usage within each period to the available amount.

The following assumptions are made in this study:

- The precedence relationships among activities are deterministic. Each activity cannot start until its predecessor activities have finished.
- The duration of each activity (d_j) is known and fixed. The quantity of capital available in each period is known and remains constant. Any remaining resources at the end of each period cannot be used in any later period.
- Once an activity is started, it cannot be interrupted. Also, preemption is not allowed.
- The cash flow (CF_j) for each activity are known. The discount rate (α) is also known. The completion time for each activity (T) is used to calculate the total project NPV.

5.3 [m-CCF] Scheduling Algorithms

m-CCF priority rule based scheduling approach consists of two components, a m-CCF priority rule to determine the list with the rankings of activities and a schedule generation scheme (SGS) to construct a feasible project schedule based on the constructed activity list.

5.3.1 [m-CCF] rule

m-CCF priority rule technique operates by dynamically selecting an activity with highest m-CCF value from a list of available activities without violating precedence, critical path and other constraints.

Due to the fact that the proposed heuristic is based on CPM and CCF, reviewing some definitions of both is necessary.

The critical path method (CPM) is very popular and it is used for scheduling a set of project activities and is widely used in computing project scheduling. CPM is a technique for managing and scheduling projects during implementation, and it can be defined as the longest path (according to the time duration) from the first node to the last node. In this method, CPM calculates the longest path of planned activities to the end of the project, and for every single task it computes the earliest and latest time a task that can start and finish without making the project longer.

Cumulative Cash Flow (CCF): Priority is given to the activity with the largest sum of cash flows for the activity and all of it successors. (This measure has been used by Baroum and Patterson [165]). This rule can be expressed as following;

Cumulative Cash Flow rule function [165]

$$CCF_j = \max(CF_j + \sum_{k \in S_j} CF_k)$$

where S_j defines the set of successors of activity j.

m-CCF is the modified version of CCF. However, rather than considering only the cash flows or the number of all follower activities, m-CCF use the discounted value of all future cash flows of successor activities. Discounting activity cash flows should better reflect the time impact of cash receipts and disbursements. Thus, this technique adds some discount factors to the undiscounted cash flow.

m-CCF value to each activity is the sum of the cash flows for that activity plus the cash flows of all the activities that must logically follow it in the project (all successor activities).

Equation (5.6) m-CCF rule function

$$m - CCF_j = \max[CF_j \exp(-\propto t_j) + \sum_{k \in S_j} CF_k \exp(-\propto (t_k - t_j))] \qquad (5.6)$$

t = EST when operate with forward scheme

t = LST when operate with backward scheme

The cash flow is discounted by the continuously compounded rate factor. There are three concepts to consider in the present value with continuous compounding formula: time value of money, present value, and continuous compounding.

Time Value of Money - The present value with continuous compounding formula relies on the concept of time value of money. Time value of money is the idea that a specific amount today is worth more than the same amount at a future date [2].

Present Value - The basic premise of present value is the time value of money.

The general formula for discounting a flow of money occurring in t years time (CF_t) to its PV (CF_0) assuming a discount rate (r) is [32]

$$NPV = CF_0 = \frac{CF_t}{(1+r)^t} = CF(1+r)^{-t}$$

The factor $(1+r)^{-t}$ is called the discount factor.

Continuous Compounding - Continuous Compounding is essentially compounding that is constant. Ordinary compounding will have a compound basis such as monthly, quarterly, semi-annually, and so forth. However, continuous compounding is nonstop, effectively having an infinite amount of compounding for a given time.

The term $\exp(-\alpha t_i)$ in equation (5.6) is nothing more than a discount factor like $(1+r)^{-t}$, except that α is continuously compounded (rather than compounded annually).

Here, in order to consider the case where the compound discount is compounded continuously, CF can be approximated using Euler's limit theorem. The approximation procedures are as follows.

From NPV formula's [32]

$$CF_t = CF_0(1+r)^t$$

Let

r = Effective discount rate

t = periods (according to r)

m = the number of times compounded

α = The discount rate (continuously compounding)

Hence, in this case, the effective discount rate (r) becomes $\frac{\propto}{m}$

and the overall compounding periods becomes mt .

NPV formula's becomes

$$CF_t = CF_0(1+\frac{\propto}{m})^{mt}$$

When m becomes infinite as it is continuously compounded.

Then mt becomes infinite large and $\frac{\propto}{m}$ infinitely small

$$CF_t = CF_o\,[\lim_{m\to\infty}\left(1+\frac{\propto}{m}\right)^{mt}]$$

In order to rearrange the equation to match with Euler's theory, let k = $\frac{m}{\propto}$ and mt = kαt

Substituting gives

$$CF_t = CF_o[\lim_{k\to\infty}\left(1+\frac{1}{k}\right)^{k}]^{\propto t}$$

From Euler's number [2]

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^{n} = 2.71828 = e$$

NPV formula's then becomes

Equation (5.7) Continuous compounding function [32]

$$CF_t = CF_0 e^{\propto t} \qquad (5.7)$$

Thus, the equation (5.7) can be re written as

$$CF_0 = CF_t e^{-\propto t}$$

which CF_0 is the present value of cash flow.

$$NPV = CF_0 = CF_t e^{-\propto t} \; or \; NPV = CF_t \exp(-\propto t)$$

The present value with continuous compounding formula is used to calculate the current value of a future amount that has earned at a continuously compounded rate (i.e. investments and loans).

5.3.2 [m-CCF] Serial Schedule Generation Scheme (SSGS)

[m-CCF]-SSGS concept is to select the activities one by one from the list and to schedule it as-soon-as-possible in the schedule.

It consists of g = 1, ..., n stages, in each of which one activity is selected and scheduled. Associated with each stage are two disjoint activity-sets: The complete set C_g contains the activities, which were already scheduled and are completed. The decision set D_g contains the unscheduled activities with every predecessor being in the complete set.

In each stage one activity from the decision set is selected with a priority rule (in case of ties the activity with the smallest activity number is selected) and scheduled at its earliest precedence and resource feasible start time.

In this scheme, the activities are selected according to the m-CCF value as follow.

Equation (5.8) m-CCF objective function for SSGS/PSGS

$$m - CCF_j = \max[CF_j \exp(-\propto EST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (EST_k - EST_j))] \qquad (5.8)$$

Priority is given to the critical activities following by the activity with the largest cash flow and its successors, where discounting based on the critical path determined early start time.

Afterwards, the selected activity is removed from the decision set and put into the complete set. This, in turn, may place a number of activities into the decision set, since all their predecessors are now scheduled. The algorithm terminates at stage number g = n, when all activities are in complete set.

To give a formal description of the SSGS some additional notation has to be introduced.

Let K_t the left over capacity of the capital resource in period t, I, the capital available, and D_g, the decision set, be defined as follows:

Equation (5.9) The left over capacity of the capital resource at period t

$$K_t = I - \sum_{j \in A_t} CF_j \qquad (5.9)$$

Equation (5.10) The Decision Set description for SSGS/B-SSGS

$$D_g = [\, j |\, j \notin C_g, P_j \subseteq C_g] \qquad (5.10)$$

Further, let EFT_j denote the earliest precedence feasible finish time of activity j within the current partial schedule and let LFT_j denote the latest precedence feasible finish time of activity j as determined by backward recursion from the upper bound of the project's makespan T. Finally, let v(j) be a priority value of activity j, $j \in D_g$.

Procedure and Flow chart for priority rule-based heuristics are listed below.

<u>STEP 1</u>: The complete set is empty and all activities are available to be scheduled.

<u>STEP 2</u>: Determine the critical activities and m-CCF value of each non critical-activity and add all activities without predecessors to the eligible available lists.
Eligible activities are defined as those activities whose predecessor activities have been scheduled (precedence feasible).

<u>STEP 3</u>: Priority is given to the critical activities then the activity with the largest cash

flow and its successors, where discounting based on the critical path determined early start time as shown below.

$$v(j) = \max[CF_j \exp(-\propto EST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (EST_k - EST_j))]$$

STEP 4: If the capital availabilities are sufficient for the duration of the task, assign the chosen task to begin at the earliest possible period, at or after its early start time. Update capital availabilities

If the capital requirements of the activity exceed the quantity of capital that are currently available, choose one of the solutions as follow.

- Delay the starts time of eligible activity to the instant at which capital become available.
- In case the capital is still exceeded after it delayed eligible activity to its latest start; balance the negative cash flow by selecting the lowest priority value of activity from C_g without violating any constraints at their original period.

STEP 5: Add any activities to the available list that become available by virtue of their predecessors being completed.

STEP 6: Repeat step 4 to 5 until all activities have been scheduled.

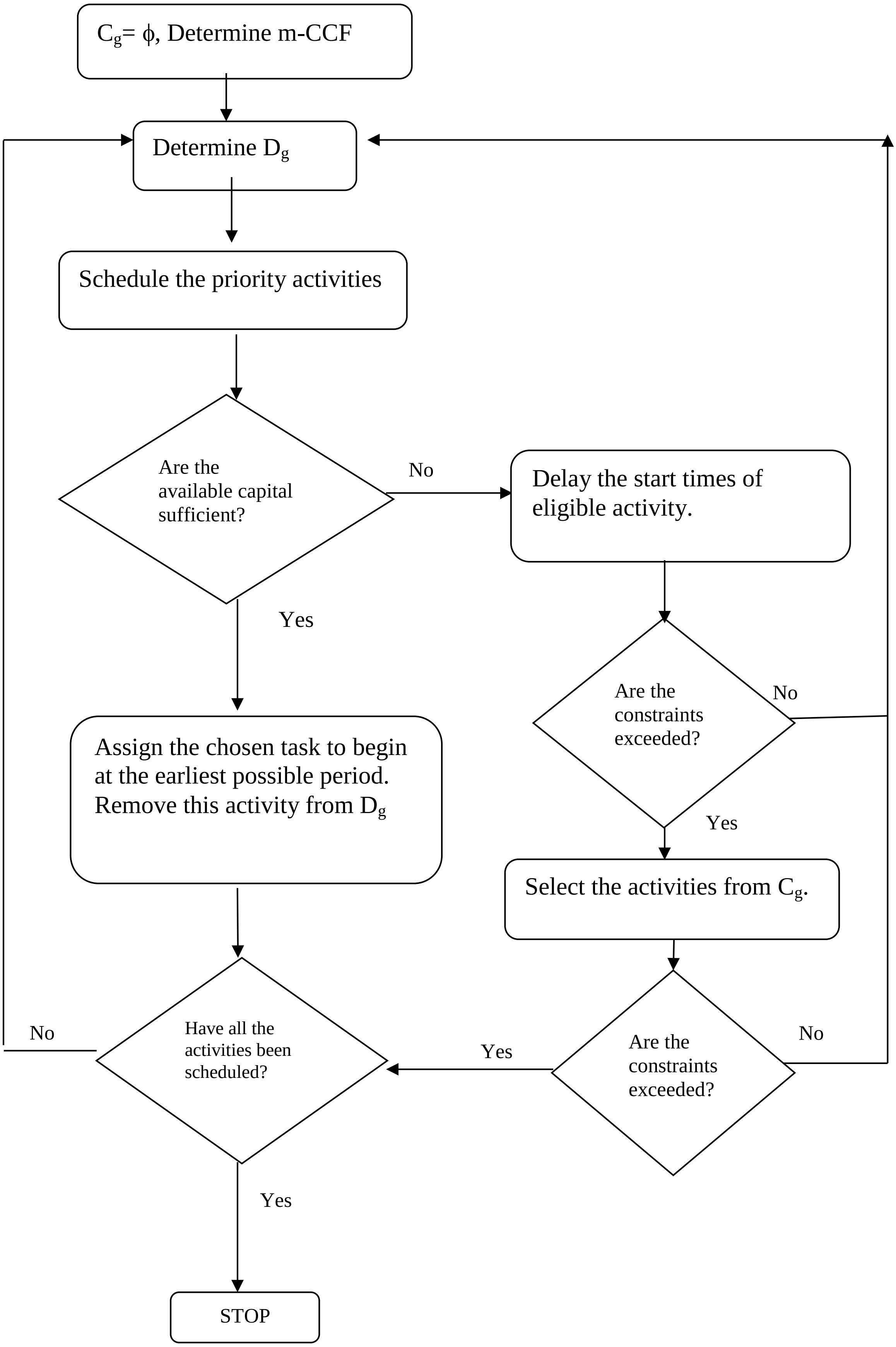


Figure 5.1: Flow chart for [m-CCF]-SSGS heuristics.

The [m-CCF]-SSGS can be formally described as follows:

Initialisation: g = 1, $C_g = \phi$; $K_t = I$;

WHILE $|C_g| < n$ DO **Stage** g

BEGIN

COMPUTE D_g and K_t, t = 1,…, T;

IF $K_t > 0$ THEN GOTO (1) ELSE GOTO (2)

(1)

Select one $j \in D_g$

$j^* = max[\, v(j) |\, j \in D_g];$

$EFT_{j^*} = max[\, FT_i |\, i \in P_j] + d_j;$

$FT_{j^*} = min[\, t |\, EFT_{j^*} \le t \le LFT_{j^*}, K_t > 0\,];$

COMPUTE K_t; IF $K_t > 0$ THEN GOTO (4) ELSE GOTO (2)

(2)

Select one $j \in C_g$

$j^* = min[\, v(j) |\, j \in C_g];$

$EFT_{j^*} = max[\, FT_i |\, i \in P_j] + d_j$

$FT_{j^*} = min[\, t |\, EFT_{j^*} \le t \le LFT_{j^*}, K_t > 0];$

COMPUTE K_t; IF $K_t > 0$ THEN GOTO (4) ELSE GOTO STEP (3)

(3)

COMPUTE C_g; IF $C_g \neq \phi$ THEN GOTO (4) ELSE STOP

(4)

$C_{g+1} = C_g \cup [j^*];$

g = g + 1;

END;

Stop

5.3.3 [m-CCF] Parallel Schedule Generation Scheme (PSGS)

[m-CCF]-PSGS concept is to select the activities at each predefined time period the activities available to be scheduled and schedules them in the list as long as constraints not exceeded.

The parallel method consists of at most J stages in each of which a set of activities (which might be empty) is scheduled. A unique feature of the parallel method is that each stage g is associated with a schedule time t_g,

On account of this schedule time, the set of scheduled activities is now divided into the following two subsets: Activities, which were scheduled and are completed up to the schedule time, are in the complete set C_g. The activities, which were scheduled and in progress i.e. still active, are in the active set A_g. In contrast to the serial method, the decision set D_g contains all yet unscheduled activities, which are available for scheduling.

The partial schedule of each stage is made up by the activities in the complete set and the active set. The schedule time of a stage equals the earliest completion time of activities in the active set of the previous stage. Each stage is made up of two steps: (1) The new schedule time is determined and activities with a finish time equal to the (new) schedule time are removed from the active set and put into the complete set. This, in turn, may place a number of activities into the decision set. (2) One activity from the decision set is selected with a priority rule (again, in case of ties the activity with the smallest label is chosen) and scheduled to start at the current schedule time.

For this scheme, the activities are selected according to the m-CCF value as follow.

$$m - CCF_j = \max[CF_j \exp(-\propto EST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (EST_k - EST_j))] \qquad (5.8)$$

Priority is given to the critical activities and the activity with the largest cash flow and its successors where discounting based on the critical path determined early start time.

Afterwards, this activity is removed from the decision set and put into the active set. Step (2) is repeated until the decision set is empty, i.e. activities were scheduled or are no longer available for scheduling. The parallel method terminates when all activities are in the complete or active set.

Given A_g, the active set, and C_g, the complete set, respectively, K_t the left over capacity of the capital resource in stage g, and D_g, the decision set, are defined as follows:

Equation (5.11) The left over capacity of the capital resource at the schedule time

$$K_g = I - \sum_{j \in A_g} CF_j \qquad (5.11)$$

Equation (5.12) The Decision Set description for PSGS/B-PSGS

$$D_g = [\, j \,|\, j \notin (C_g \cup A_g), P_j \subseteq C_g] \qquad (5.12)$$

Procedure and Flow chart for priority rule-based heuristics are listed below.

STEP 1: The complete set is empty and all activities are available to be scheduled.

STEP 2: Determine the m-CCF value of each activity and add all activities without predecessors to the available lists.

STEP 3: Priority is given to the critical activities and the activity with the largest cash flow and its successors where discounting based on the critical path determined early start time as shown below.

$$v(j) = \max[CF_j \exp(-\propto EST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (EST_k - EST_j))]$$

STEP 4: Determine the schedule time of a stage (earliest start time). Except from the first stage, the schedule time equals the earliest completion time of activities in the active set of the previous stage.

STEP 5: Schedule eligible activities with the use of a priority rule to begin at the schedule time until the decision set in this stage is empty.

If the capital requirements of the activity exceed the quantity of capital that is currently available, choose one of the solutions as follow.

- Eligible activities that were not scheduled in this stage remain in the decision set with their start times delayed until the next schedule time or when the capital become available.

- In case the capital is still exceeded after delaying all eligible activities, balance the cash outflow by selecting the lowest priority value of activity from C_g without violating any constraints at their original period. Adjust the start time.

STEP 6: Precedence feasible activities whose start times are current at that time and their predecessors being completed are then added to the queue of eligible activities. This step examines the decision set and new schedule time.

STEP 7: Repeat step 4 to 6 until all activities have been scheduled.

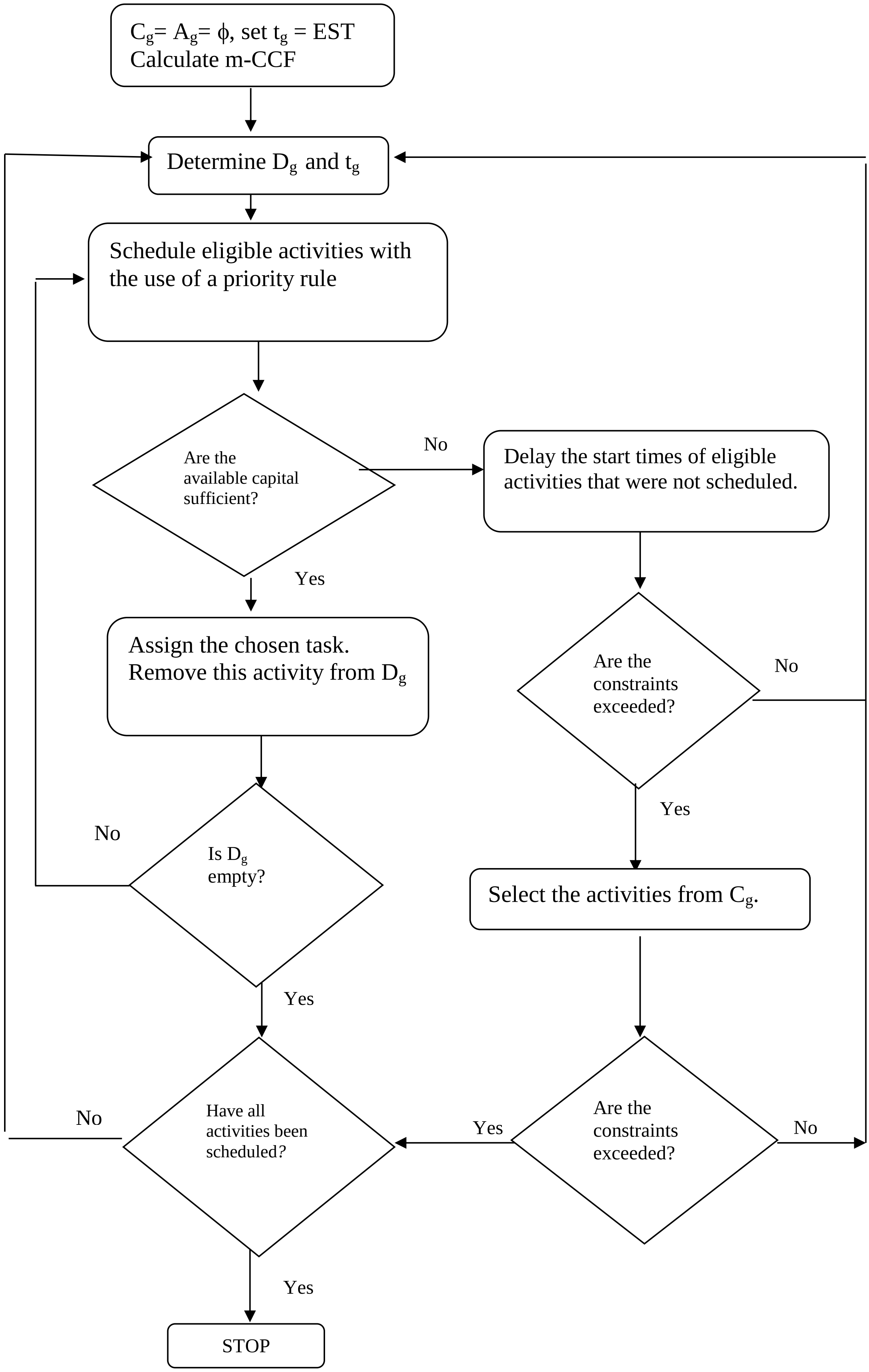


Figure 5.2: Flow chart for [m-CCF]-PSGS heuristics.

The [m-CCF]-PSGS can be formally described as follows:

Initialization: g = 1, t_g= 0, $A_g = C_g = \phi$, D_g = [1], K_g = I

WHILE $|A_g \cup C_g| < n$ DO **Stage** g

BEGIN

(1)

$t_g = min[FT_j | j \in A_{g-1}]$;

$A_g = A_{g-1} \setminus [j | j \in A_{g-1}, FT_j = t_g]$;

$C_g = C_{g-1} \cup [j | j \in A_{g-1}, FT_j = t_g]$;

COMPUTE K_g ; IF $K_g > 0$ THEN GO TO STEP (2) ELSE GOTO STEP (3);

(2)

Select one $j \in D_g$

$j^* = max[v(j) | j \in D_g]$;

$FT_{j^*} = t_g + d_{j^*}$;

COMPUTE K_g ; IF $K_g > 0$ THEN GO TO STEP (5) ELSE GOTO STEP (3);

(3)

Select one $j \in C_g$

$j^* = min[v(j) | j \in C_g]$; ()

$FT_{j^*} = t_g + d_{j^*}$;

COMPUTE K_g; IF $K_g > 0$ THEN GO TO STEP (5) ELSE GOTO STEP (4) ;

(4)

COMPUTE C_g ; IF $C_g \neq \phi$ THEN GOTO (5) ELSE STOP

(5)

$A_g = A_g \cup [j^*]$;

COMPUTE D_g; IF $D_g \neq \phi$ THEN GO TO STEP (2) ELSE g = g +1;

END;

Stop

5.4 Backward Strategy

The serial and parallel scheduling schemes for constructing feasible schedules are extended by the flexible use of different planning directions (forward and backward).

The backward planning strategies (can be referred to as late start scheme) are incorporated into priority rule-based procedures. This idea has been introduced by Tormos and Lova [179] and also used by Klein [189], Li and Willis [115], and Ozdamar and Ulusoy [116].

Within backward SSGS, a backward available job is selected in each iteration and scheduled as late as possible without violating the precedence constraints [114]. Usually, the schedule obtained will not start at the beginning of the planning horizon.

5.4.1 [m-CCF] Backward-SSGS (B-SSGS)

[m-CCF]-B-SSGS concept is to select the activities one by one from the list and schedules it as-late-as-possible in the schedule.

It consists of g = 1, ..., n stages, in each of which one activity is selected and scheduled. Associated with each stage are two disjoint activity-sets: The complete set C_g is the activities, which were already scheduled and are completed. The decision set D_g contains the unscheduled activities with every predecessor being in the complete set.

In each stage one activity from the decision set is selected with a priority rule (in case of ties the activity with the smallest activity number is selected) and scheduled at its earliest precedence and resource feasible start time.

In this scheme, the activities are selected according to the m-CCF value as follow.

Equation (5.13) m-CCF objective function for B-SSGS/B-PSGS

$$m - CCF_j = \max[CF_j \exp(-\propto LST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (LST_k - LST_j))] \qquad (5.13)$$

Priority is given to the activity with the largest cash flow and its successors, where discounting based on the critical path determined late start time.

Afterwards, the selected activity is removed from the decision set and put into the complete set. This, in turn, may place a number of activities into the decision set, since all their predecessors are now scheduled. The algorithm terminates at stage number g = n, when all activities are in the complete set.

A formal description of the backward serial scheduling scheme is the same as in forward strategy. Let K_t the left over capacity of the capital resource in period t, and D_g, the decision set, be defined as follows:

$$K_t = I - \sum_{j \in A_t} CF_j \qquad (5.9)$$

$$D_g = [\, j |\, j \notin C_g, P_j \subseteq C_g] \qquad (5.10)$$

Procedure and Flow chart for B-SSGS priority rule-based heuristics are listed below.

STEP 1: The complete set (C_g) is empty and all activities are available to be scheduled.

STEP 2: Determine the m-CCF value of each non-critical activity and add all activities without predecessors to the available lists.

STEP 3: Priority is given to the critical activities and the activity with the largest cash flow and its successors, where discounting based on the critical path determined late start time as shown below.

$$v(j) = \max[CF_j \exp(-\propto LST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (LST_k - LST_j))]$$

STEP 4: If the capital availabilities are sufficient for the duration of the task, assign the chosen task to begin at the latest possible period, at or before its late start time. Update capital availabilities.

If the capital requirements of the activity exceed the quantity of capital that is currently available, choose one of the solutions as follow.

- Move activity forward until the capital requirement can be fulfilled.
- In case the capital is still exceeded after shifting the activity to its earliest start time, move the lowest priority value of activity from C_g until the requirement become fulfill.

STEP 5: Add any activities to the available list that become available by virtue of their predecessors being completed. This procedure examines activities and updates D_g.

STEP 6: Repeat step 4 to 5 until all activities have been scheduled.

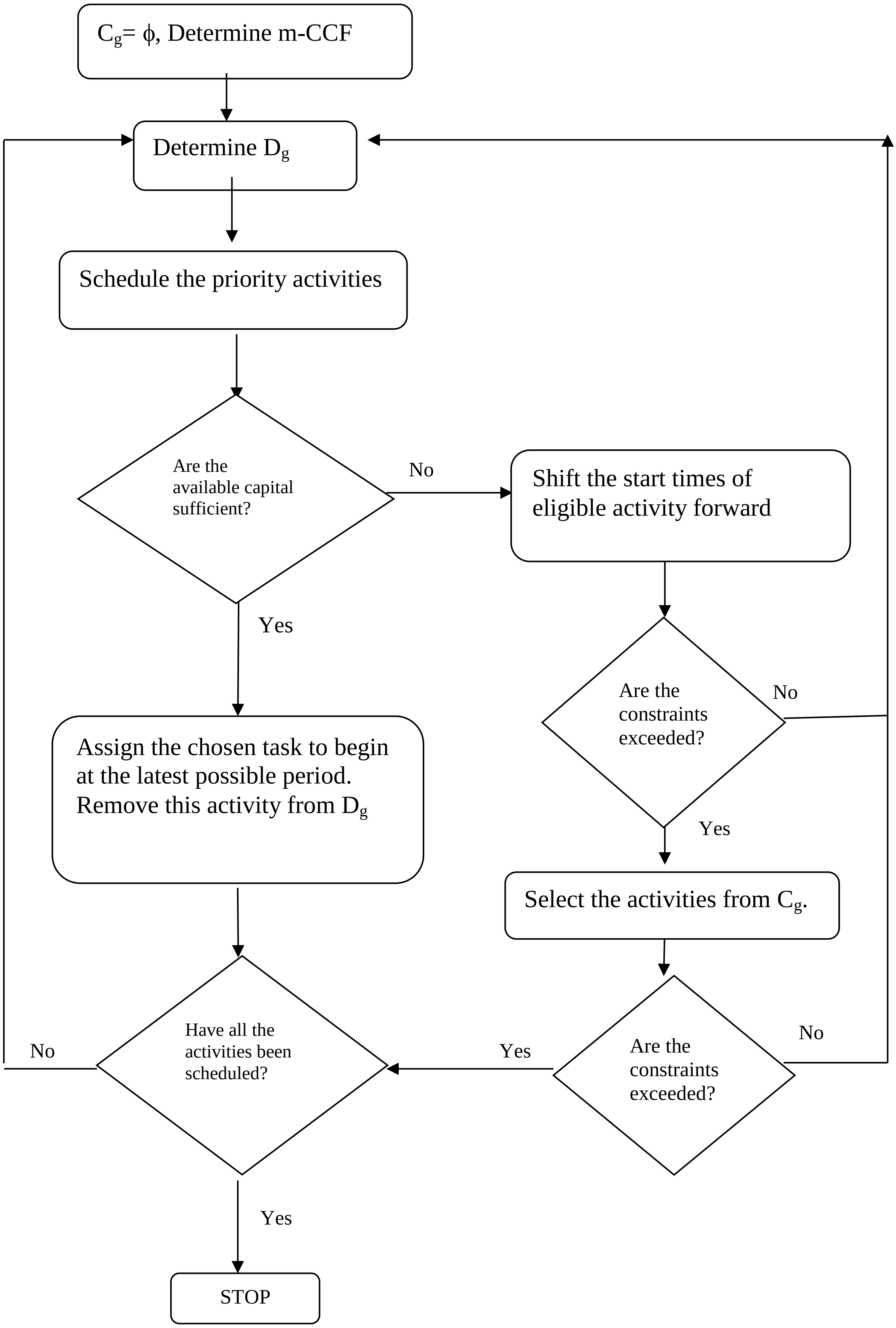


Figure 5.3: Flow chart for [m-CCF]-B-SSGS heuristics.

The [m-CCF]-B-SSGS can be formally described as follows:

Initialisation: g = 1, $C_g = \phi$;

WHILE $|C_g| < n$ DO **Stage** g

BEGIN

COMPUTE D_g and K_t, t = 1,..., T;

IF $K_t > 0$ THEN GOTO (1) ELSE GOTO (2)

(1)

Select one $j \in D_g$

$j^* = max[\, v(j) |\, j \in D_g]$;

$EFT_{j^*} = max[\, FT_i |\, i \in P_j] + d_j$;

$FT_{j^*} = max[\, t |\, EFT_{j^*} \leq t \leq LFT_{j^*},\ K_t > 0]$;

COMPUTE K_t; IF $K_t > 0$ THEN GOTO (4) ELSE GOTO (2)

(2)

Select one j □ C_g

$j^* = min[\, v(j) |\, j \in C_g]$;

$EFT_{j^*} = max[\, FT_i |\, i \in P_j] + d_j$

$FT_{j^*} = max[\, t |\, EFT_{j^*} \leq t \leq LFT_{j^*},\ K_t > 0]$;

COMPUTE K_t; IF $K_t > 0$ THEN GOTO (4) ELSE GOTO (3)

(3)

COMPUTE C_g; IF $C_g \neq \phi$ THEN GOTO (4) ELSE STOP

(4)

$C_{g+1} = C_g \cup [j^*]$;

g = g + 1;

END;

Stop

5.4.2 [m-CCF] Backward-PSGS (B-PSGS)

The partial schedule of each stage is made up by the activities in the complete set and the active set. The schedule time of a stage equals the latest completion time of activities in the active set of the previous stage. Each stage is made up of two steps: (1) The new schedule time is determined and activities with a finish time equal to the (new) schedule time are removed from the active set and put into the complete set. This, in turn, may place a number of activities into the decision set. (2) One activity from the decision set is selected with a priority rule (again, in case of ties the activity with the smallest label is chosen) and scheduled to start at the current schedule time.

For this scheme, the activities are selected based on the m-CCF value as follow.

$$m - CCF_j = \max[CF_j \exp(-\propto LST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (LST_k - LST_j))] \qquad (5.13)$$

Priority is given to the activity with the largest cash flow and its successors, where discounting based on the critical path determined late start time.

Afterwards, this activity is removed from the decision set and put into the active set. Step (2) is repeated until the decision set is empty, i.e. activities were scheduled or are no longer available for scheduling. The parallel method terminates when all activities are in the complete or active set.

Given A_g, the active set, and C_g, the complete set, respectively, K_t the left over capacity of the capital resource, and D_g, the decision set, are defined as follows:

$$K_g = I - \sum_{j \in A_g} CF_j \qquad (5.11)$$

$$D_g = [\, j |\, j \notin (C_g \cup A_g), P_j \subseteq C_g] \qquad (5.12)$$

Procedure and Flow chart for B-PSGS priority rule-based heuristics are listed below.

STEP 1: The complete set is empty and all activities are available to be scheduled.

STEP 2: Determine the m-CCF value of each activity and add all activities without predecessors to the available lists.

STEP 3: Priority is given to the critical activities and the activity with the largest cash flow and its successors, where discounting based on the critical path determined late start time as shown below.

$$v(j) = \max[CF_j \exp(-\propto LST_j) + \sum_{k \in S_j} CF_k \exp(-\propto (LST_k - LST_j))]$$

STEP 4: Determine the schedule time of a stage (latest start time). Except from the first stage, the schedule time equals the latest completion time of activities in the active set of the previous stage.

STEP 5: Schedule eligible activities with the use of a priority rule until the decision set in this stage is empty.

If the capital requirements of the activity exceed the quantity of capital that is currently available, choose one of the solutions as follow.

- Move activities that were not scheduled forward until the capital requirement can be fulfilled. Adjust the schedule time.

- In case the capital is still exceeded after shifting the activity to its earliest start time, move the lowest priority value of activity from C_g until all the requirement become fulfill. Adjust the schedule time.

STEP 6: Precedence feasible activities whose start times are current at that time and their predecessors being completed are then added to the queue of eligible activities. This step examines the decision set and new schedule time.

STEP 7: Repeat step 4-6 until all activities have been scheduled.

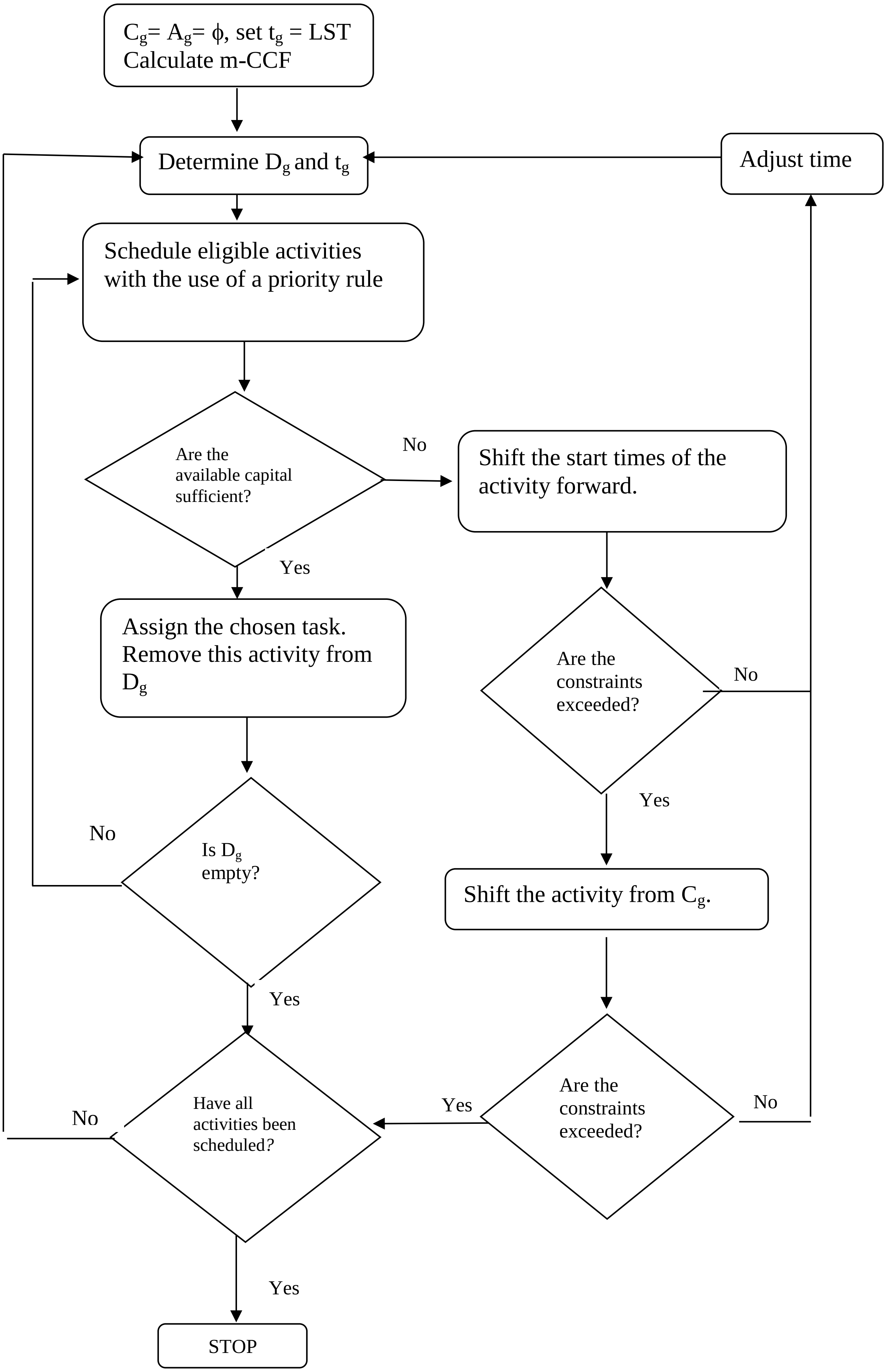


Figure 5.4: Flow chart for [m-CCF]-B-PSGS heuristics.

The [m-CCF]-B-PSGS can be formally described as follows:

Initialization: g = 1, t_g= 0, $A_g = C_g = \phi$, D_g = [1], K_g = I

WHILE $|A_g \cup C_g| < n$ DO **Stage** g

BEGIN

(1)

$t_g = max[\, FT_j |\, j \in A_{g-1}];$

$A_g = A_{g-1} \setminus [\, j |\, j \in A_{g-1}, FT_j = t_g];$

$C_g = C_{g-1} \cup [\, j |\, j \in A_{g-1}, FT_j = t_g];$

COMPUTE K_g ; IF $K_g > 0$ THEN GO TO STEP (2) ELSE GOTO STEP (3);

(2)

Select one $j \in D_g$

$j^* = max[\, v(j) |\, j \in D_g];$

$FT_{j^*} = t_g + d_{j^*};$

COMPUTE K_g ; IF $K_g > 0$ THEN GO TO STEP (5) ELSE GOTO STEP (3);

(3)

Select one $j \in C_g$

$j^* = min[\, v(j) |\, j \in C_g];$

$FT_{j^*} = t_g + d_{j^*};$

COMPUTE K_g; IF $K_g > 0$ THEN GO TO STEP (5) ELSE GOTO STEP(4) ;

(4)

COMPUTE C_g; IF $C_g \neq \phi$ THEN GO TO STEP (5) ELSE STOP;

(5)

$A_g = A_g \cup [j^*];$

COMPUTE D_g; IF $D_g \neq \phi$ THEN GO TO STEP (2) ELSE g = g +1;

END;

Stop

5.5 Experimental design

The experiment is conducted in three phases. In the first phase, the proposed m-CCF heuristic solution is compared with other rules which are MINSLK, GNS and CCF. Only the smaller test problems are included in this phase, since solving a very large-scale problem would require a very large amount of computational effort. This comparison is made by using Network Diagram adapted from Baroum's Example [165]. The superiority and applicability of the m-CCF method for larger test problems are confirmed further by performing additional computations and comparisons in the next two phases. In the second phase, m-CCF method is implemented in two scheme (serial and parallel) and two strategies (forward and backward), then it is compared with the optimal solution of Doersch and Patterson [162]. In the third phase, the main purpose is to address an area of strength and weakness in the chosen technique. The m-CCF is test in three different dataset.

As mentioned earlier, the data are collected from both literature and PTT Public company limited during the author's fieldwork in Thailand. All of the scheduling rules and schemes of the proposed technique are coded in MATLAB (R2011a). Solver tools of the latest version of Excel spreadsheet are used to analyse the research data collected and facilitate searching the optimal solution.

5.5.1 The First Phase

In developing the scheduling heuristics to maximise project NPV, Priority-based heuristics are constructed from the parameters CF, α and t. The proposed m-CCF heuristic solution is compared with other rules which are MINSLK, GNS and CCF.

Three rules (below) based on these activity parameters are presented. Each technique operates by dynamically selecting an activity with highest important from a list of available activities without violating precedence, critical path and other constraints.

Minimum slack (MINSLK) priority rule is one of the first standard heuristic scheduling rules. Priority in resolving resource conflicts is given to the activity with minimum slack or total float, where activity slack is based upon the traditional critical path (non-resource constrained) solution [190]. (The highest rated rule for minimise project duration in (Davis and Patterson [9] and Russell [169]). This rule can be expressed as following;

Equation (5.14) MINSLK rule function [190]

$$MINSLK_j = \min(LFT_j - EFT_j) \qquad (5.14)$$

Greatest Number of Successors (GNS): Priority is assigned to activities with the greatest number of successors [190]. This rule can be expressed as following;

Equation (5.15) GNS rule function [190]

$$GNS_j = \max \sum_{k \in S_j} I_k \qquad (5.15)$$

where I_k be a number of successors of activity j.

Cumulative Cash Flow weight (CCF): Priority is given to the activity with the largest sum of cash flows for the activity and all of it successors. (This measure has been used by Baroum and Patterson [165]. This rule can be expressed as following;

Equation (5.16) CCF rule function [165]

$$CCF_j = \max(CF_j + \sum_{k \in S_j} CF_k) \qquad (5.16)$$

where S_j defines the set of successors of activity j

5.5.2 The Second Phase

In the second phase, m-CCF method is implemented in two schemes (serial and parallel) and two strategies (forward and backward), and then it is compared with the optimal solution.

The optimal solution presented in this section is based upon the binary integer programming formulation of Doersch and Patterson [162] that have been discussed on Chapter 4. Unfortunately, this technique can only solve unrealistically small problems of marginal practical value.

The terms in the objective function represent cash outflows, cash inflows and capital costs, respectively, where each component is discounted back to the beginning of the project:

$$Maximise \sum_{k=1}^{m} CF_{i(k)} \exp\left(-\propto T_{i(k)}\right) + CF_{j(k)} \exp\left(-\propto T_{j(k)}\right)$$

$$+[I_k \exp(-\propto d_k) - I_k]\exp(-\propto T_{i(k)})$$

$$(4.4)$$

The solution from equation (4.4) is solved with GRG optimisation routine and used as an optimal solution in this research. All the problems are solved using the Solver tools of Microsoft office Excel to facilitate searching the optimal solution.

5.5.3The Third Phase

In this section, the m-CCF rule is test with different data set. These problems consist of projects with between 100 and 300 activities where the project data set are obtained from the PTT Company limited in Thailand in which the author has got accessed to during the Industrial placement periods. All three problems are solved with each of the proposed heuristics procedure described, [m-CCF]-SSPS, -PSGS, -B-SSPS and B--PSGS then compare them with the actual NPV of this projects. This is done in order to assess the efficacy of each of the heuristic scheduling rules under different cash flow patterns.

Figure 5.5 displays all the three phases experiment that are carried out in this study. The first phase of experiment is presented in Chapter 6 and the last two phases are reported in Chapter 7.

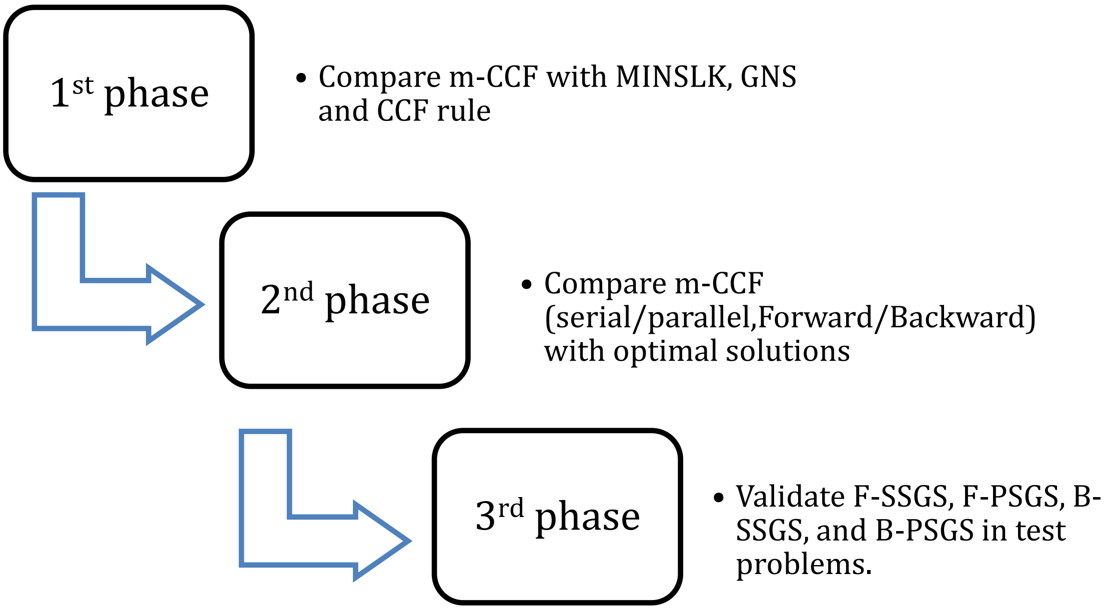


Figure 5.5 Experimental design in this study

5.6 Summary

This chapter has presented the basic assumptions and concepts using in this study. The chapter continues with the description of m-CCF technique and the scheme used for schedule all the activities. The experiment in later chapters is conducted in three phases. The first phase is to test the new heuristic algorithm rule and compare its results with the previous rule from literature. On second phase, the rule is measured by employing on problems in both serial and parallel scheme. The last phase, the effectiveness of the heuristic rule is validated through three different projects. An area of strength and weakness in the chosen technique is identified.

The next chapter will present the results of m-CCF technique and compare it with other techniques.

Chapter 6 Results I: A Review of algorithm performance

6.1 Introduction

This chapter contains the first phase experiment that has been conducted to test the effectiveness of the proposed heuristic rule, m-CCF. The proposed priority rule based heuristic has been evaluated and compared with other three rules MINSLK, GNS and CCF respectively. The output of each sample schedule, which was provided by various algorithms in existing literatures, is compared with the m-CCF rule output. Some data reported in this chapter have been taken and adapted from author's own research [191].

6.2 The First Phase Experiment

The proposed m-CCF heuristic solution is compared with other rules which are MINSLK, GNS and CCF. Each technique operates by dynamically selecting an activity with highest important from a list of available activities without violating precedence, critical path and other constraints.

6.2.1 MINSLK and m-CCF

Network Diagram adapted from Russell's [169]. The activities in red denote the activities in critical path. Full numerical illustration can be seen on Appendix A.

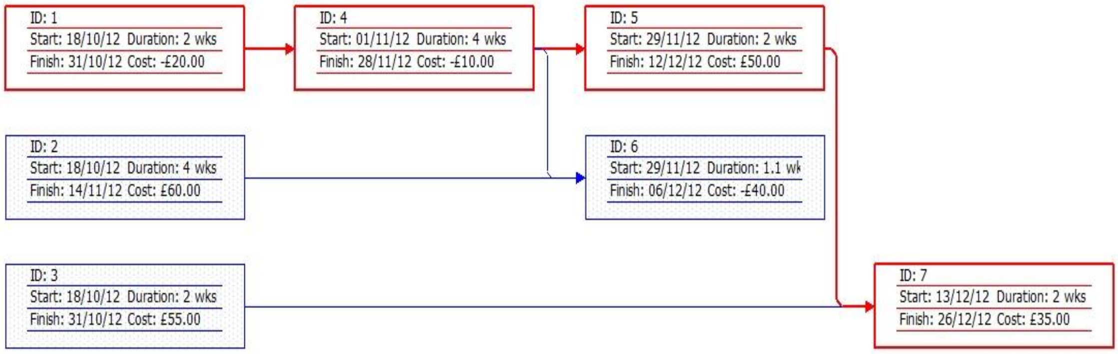


Figure 6.1 Network Diagrams for MINSLK and m-CCF comparison

According to [m-CCF]-SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown Table 6.1.

Table 6.1 m-CCF values for each activity

Task						m-CCF
1	-20	-9.61	44.35	-34.08	29.82	10.47
2	60	-34.08				25.91
3	55	32.31				87.31
4	-10	46.15	-35.48	31.04		31.72
5	50	33.63				83.63
6	-40					-40
7	35					35

From equation (5.6)

$$m - CCF_j = \max[CF_j \exp(-\propto t_j) + \sum_{k \in S_j} CF_k \exp(-\propto (t_k - t_j))] \qquad (5.6)$$

Demonstration

[1] = -20 + (-10)exp(-2α) + 50 exp(-6α) + (-40) exp(-8α) + 35exp(-8α)

[2] = 60 + (-40)exp(-8α)

[3] = 55 + 35exp(-4α)

[4] = $-10 + 50\exp(-4\alpha) + (-40)\exp(-6\alpha) + 35\exp(-6\alpha)$

[5] = $50 + 35\exp(-2\alpha)$

[6] = -40

[7] = 35

The schedules after applying both rules in serial generation scheme (SSGS) are displayed in Table 6.2 and 6.3.

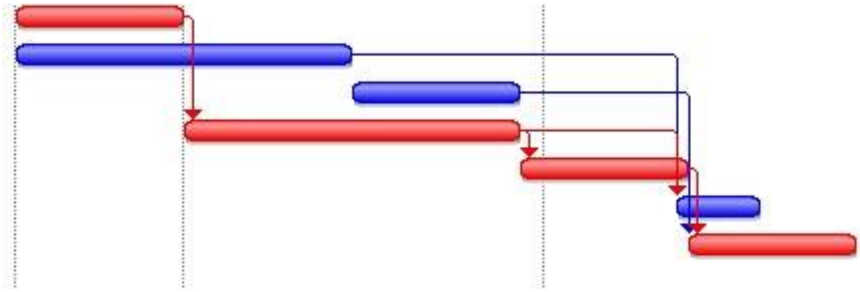
Table 6.2 The output from applying MINSLK rule

g	D_g	j
1	[1,2,3]	1,2
2	[3,4]	4
3	[3]	3
4	[5,6]	5
5	[6,7]	6,7

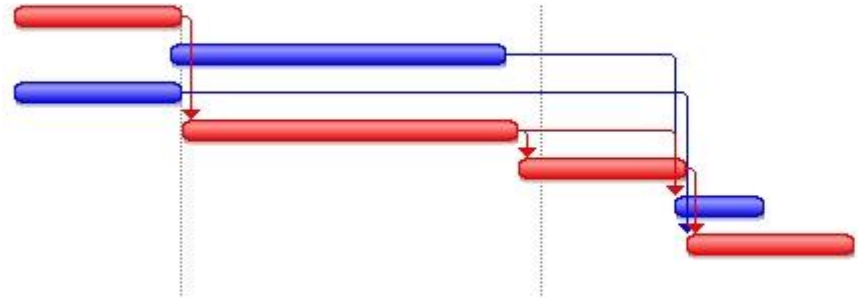
Table 6.3 The output from applying m-CCF rule

g	D_g	j
1	[1,2,3]	1,3
2	[2,4]	2,4
3	[5]	5
4	[6,7]	6,7

MINSLK put the activities in an increasing order of their slack value in the list that results in the schedule as shown in Figure 6.2(a). m-CCF put the activities in an increasing order of their m-CCF value (display in Table 6.1) that results in the schedule as shown in Figure 6.2(b).



(a) MINSLK rule



(b) m-CCF rule

Figure 6.2 The schedule from (a) MINSLK rule (b) m-CCF rule

Among the three non-critical activities, in this case 2, 3, and 6, MINSLK give priority to activity 6, 2 and 3. m-CCF gives priority to 3, 2 and 6 respectively. NPV are then calculated according to each rule as shown in Table 6.4 and 6.5.

Table 6.4 NPV obtained from MINSLK rule

Activity	NewTs	t_j	PV
1	1	0	-20
2	1	0	60
3	5	4	50.77
4	3	2	-9.61
5	7	6	44.35
6	9	8	-34.08
7	9	8	29.82
		NPV	121.25

NPV = -20+60+55exp(-4α)+(-10)exp(-2α) + 50exp(-6α)+(-40) exp(-8α) + 35exp(-8α)

NPV obtained from MINSLK rule = 121.25.

Table 6.5 NPV obtained from m-CCF rule

Activity	New Ts	t_j	PV
1	1	0	-20
2	3	2	57.65
3	1	0	55
4	3	2	-9.61
5	7	6	44.35
6	9	8	-34.08
7	9	8	29.83
		NPV	123.12

NPV = -20+60exp(-2α)+55+(-10)exp(-2α)+50exp(-6α) +(-40) exp(-8α) + 35exp(-8α)

NPV obtained from m-CFF rule = 123.12 which is increased by 1.52 %.

6.2.2 GNS and m-CCF

Network diagram is adapted from Padman's example [170]. The activities in red denote the activities in critical path. Full numerical illustration can be seen on Appendix B.

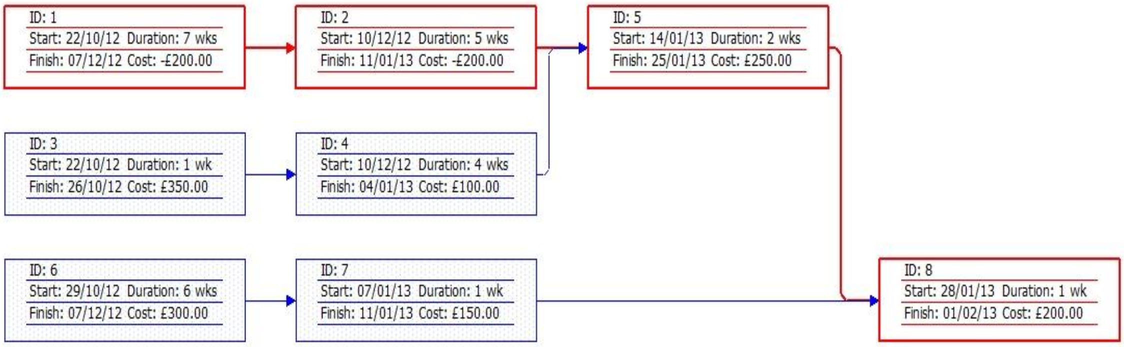


Figure 6.3 Network diagram for GNS and m-CCF comparison.

According to [m-CCF]-SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown in Table 6.6.

Table 6.6 m-CCF values for each activity

Task					m-CCF
1	-200	-173.87	196.66	151.16	-26.06
2	-200	226.21	173.87		200.08
3	350	98.02	196.66	151.16	795.83
4	100	200.63	154.21		454.84
5	250	192.16			442.16
6	300	133.038	151.16		584.19
7	150	170.43			320.43
8	200	0			200

Demonstration

[1] = $-200 + (-200)\exp(-7\alpha) + 250\exp(-12\alpha) + 200\exp(-14\alpha)$

[2] = $-200 + 250\exp(-5\alpha) + 200\exp(-7\alpha)$

[3] = $350 + 100\exp(-\alpha) + 250\exp(-12\alpha) + 200\exp(-14\alpha)$

[4] = $100 + 250\exp(-11\alpha) + 200\exp(-13\alpha)$

[5] = $250 + 200\exp(-2\alpha)$

[6] = $300 + 150\exp(-6\alpha) + 200\exp(-14\alpha)$

[7] = $150 + 200\exp(-8\alpha)$

[8] = 200

The schedules after applying both rules in SSGS are shown in Table 6.7 and 6.8.

Table 6.7 The output from applying GNS rule

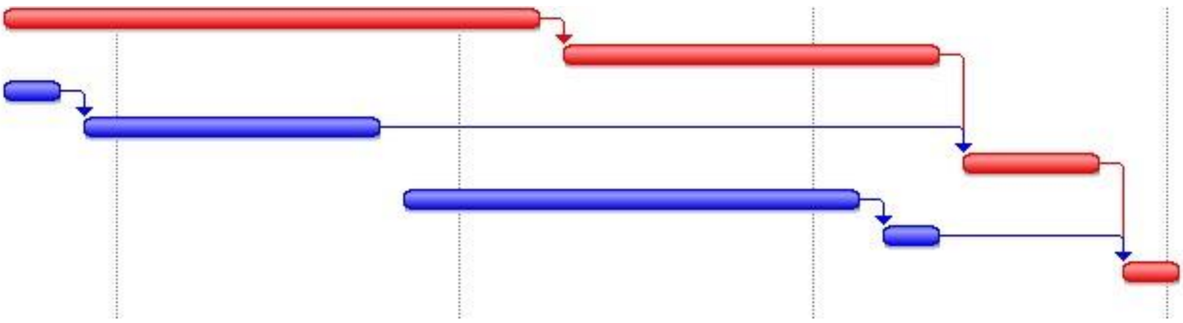
g	D_g	j
1	[1,3,6]	1,3
2	[4,6]	4
3	[6]	6
4	[2,7]	2
5	[7]	7
6	[5]	5
7	[8]	8

Table 6.8 The output from applying m-CCF rule

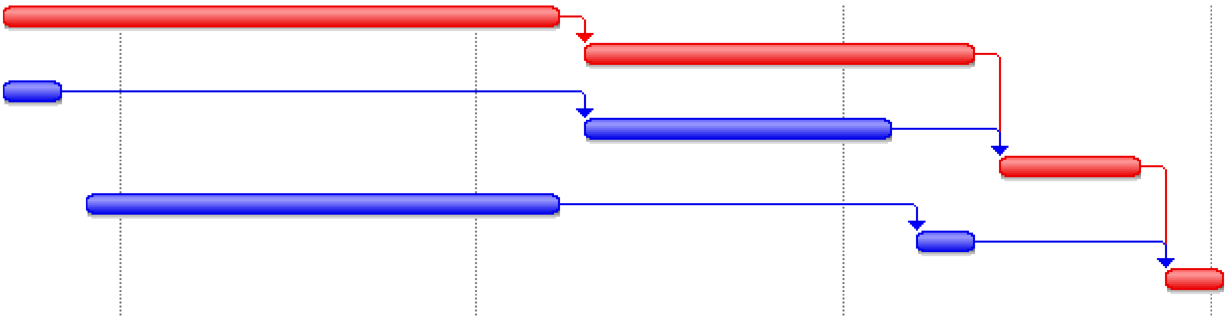
g	D_g	j
1	[1,3,6]	1,3
2	[4,6]	6
3	[2,4,7]	2,4
4	[4]	7
5	[5]	5
6	[8]	8

GNS put the activities in an increasing order of their number of successors in the schedule as shown in Figure 6.4(a).

m-CCF put the activities in an increasing order of their m-CCF value (display in Table 6.6) that results in the activity list as shown in Figure 6.4(b).



(a) GNS rule



(b) m-CCF results

Figure 6.4 The schedule from (a) GNS rule (b) m-CCF rule

Among the four non-critical activities, in this case 3, 4, 6 and 7, GNS gives priority to activity 3, 4, 6 and 7 respectively. m-CCF gives priority to 3, 6, 4 and 7 respectively.

NPV are then calculated according to each rule as shown in Table 6.9 and 6.10.

Table 6.9 NPV obtained from GNS rule

Activity	NewTs	t_j	PV
1	1	0	-200
2	8	7	-173.87
3	1	0	350
4	2	1	98.02
5	13	12	196.66
6	6	5	271.45
7	12	11	120.38
8	15	14	151.16
		NPV	662.63

NPV $= -200+(-200)\exp(-7\alpha)+350+100\exp(-\alpha)+250\exp(-12\alpha)+300\exp(-5\alpha) + 150\exp(-11\alpha)+ 200\exp(-14\alpha)$

NPV obtained from GNS rule = 662.63.

Table 6.10 NPV obtained from m-CCF rule

Activity	New Ts	t_j	PV
1	1	0	-200
2	8	7	-173.87
3	1	0	350
4	9	7	86.93
5	13	12	196.66
6	2	1	294.06
7	8	8	127.82
8	15	14	151.16
		NPV	832.76

NPV $= -200+(-200)\exp(-7\alpha)+350+100\exp(-7\alpha)+250\exp(-12\alpha)+300\exp(-\alpha) + 150\exp(-8\alpha)+ 200\exp(-14\alpha)$

NPV obtained from m-CFF rule = 832.76 which is increased by 20.43 %.

6.2.3 CCF and m-CCF

Network Diagram is adapted from Baroum's Example [165]. The activities in red denote the activities in critical path. Full numerical illustration can be seen on Appendix C.

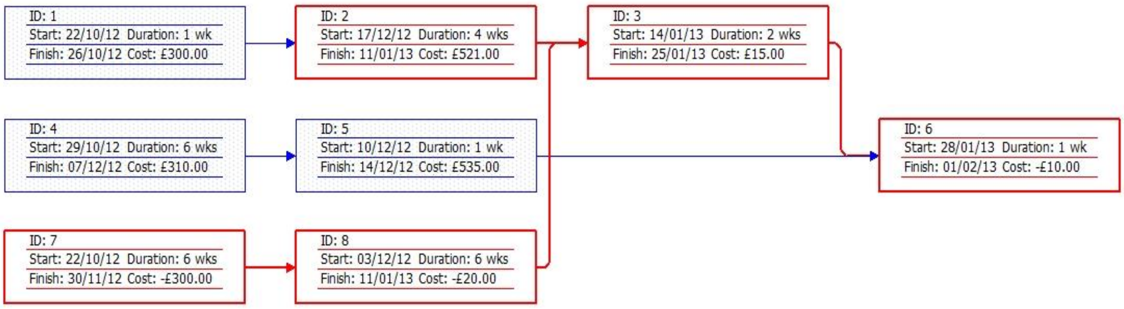


Figure 6.5 Network diagram for CCF and m-CCF comparison

From equation (5.5), the CCF values are determined as shown in Table 6.11

Table 6.11 CCF values obtained for each activity

Task	Cash Flow				CCF
1	300	521	15	-10	826
2	521	15	-10		526
3	15	-10			5
4	310	535	-10		835
5	535	-10			525
6	-10				-10
7	-300	15	-10	-20	-315
8	-20	15	-10		-15

From equation (5.16)

$$CCF_j = \max(CF_j + \sum_{k \in S_j} CF_k) \qquad (5.16)$$

Demonstration

[1] = 300+521+15+(-10) = 826

[2] = 521+15+(-10) = 526

[3] = 15+(-10) = 5

[4] = 310+535+(-10) = 835

[5] = 535+(-10) = 525

[6] = -10

[7] = -300+15+(-10)+(-20) = -315

[8] = -20+15+(-10) = -15

According to m-CCF SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown in Table 6.12.

Table 6.12 m-CCF values obtained for each activity

Task					mCCF
1	300	500.57	11.79	-7.56	804.81
2	521	12.281	-7.86		525.41
3	15	-9.61			5.39
4	310	474.50	-7.56		776.94
5	535	-8.52			526.48
6	-10				-10
7	-300	11.79	-7.56	-17.74	-313.49
8	-20	13.30	-8.52		-15.22

Demonstration

[1] = $300 + 521\exp(-2\alpha) + 15\exp(-12\alpha) + (-10)\exp(-14\alpha)$

[2] = $521 + 15\exp(-10\alpha) + (-10)\exp(-12\alpha)$

[3] = $15 + (-10)\exp(-2\alpha)$

[4] = $310 + 535\exp(-6\alpha) + (-10)\exp(-14\alpha)$

[5] = $535 + (-10)\exp(-8\alpha)$

[6] = -10

[7] = $-300 + 15\exp(-12\alpha) + (-10)\exp(-14\alpha) + (-20)\exp(-6\alpha)$

[8] = $-20 + 15\exp(-6\alpha) + (-10)\exp(-8\alpha)$

The schedules after applying both rules in SSGS coded are shown in Table 6.13 and 6.14.

Table 6.13 The output from applying CCF rule

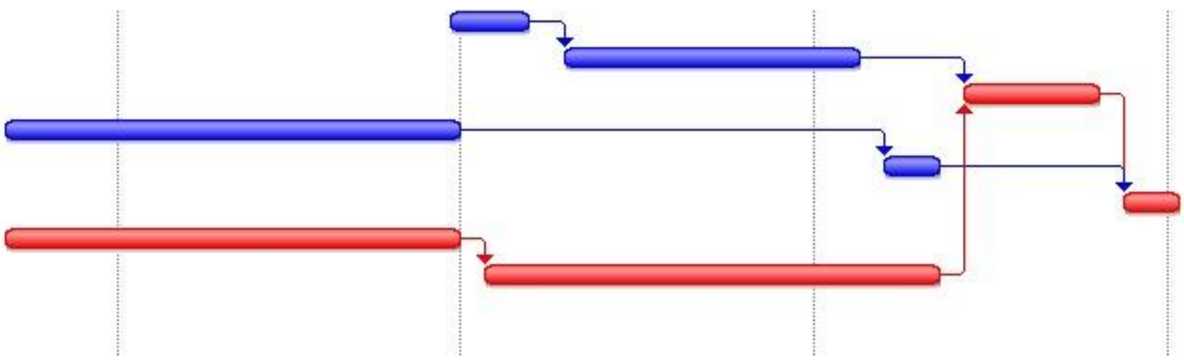
g	D_g	j
1	[1,4,7]	4,7
2	[1,5,8]	1,8
3	[2,5]	2
4	[5]	5
5	[3]	3
6	[6]	6

Table 6.14 The output from applying m-CCF rule

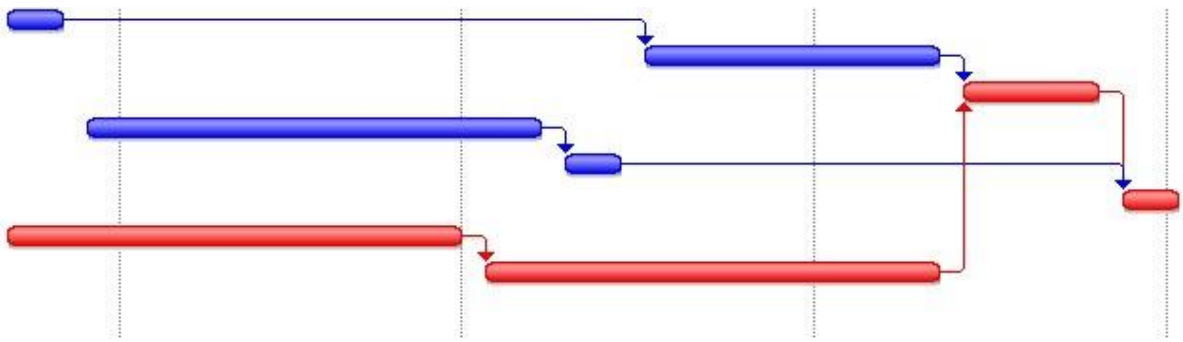
g	D_g	j
1	[1,4,7]	1,7
2	[2,4]	4
3	[2,5,8]	8
4	[2,5]	5
5	[2]	2
6	[6]	6
7	[8]	8

CCF puts the activities in an increasing order of their CCF value (display in Table 6.11) that results in the schedule as shown in Figure 6.6(a).

m-CCF puts the activities in an increasing order of their m-CCF value (display in Table 6.12) that results in the schedule as shown in Figure 6.6(b).



(a) CCF rule



(b) m-CCF rule

Figure 6.6 The schedule from (a) CCF rule (b) m-CCF rule

In this case, the non-critical activities are 1, 2, 4,and 5.

CCF give priority to activity 4, 1, 2 and 5. Thus they schedule activity 4 first by shift to the right as much as possible followed by activity 1, 2 and 5.

m-CCF gives priority to 1, 4, 5 and 2 respectively. Thus they schedule activity 1 first by shifting to the right as much as possible followed by activity 4 and 5.

NPV are then calculated according to each rule as shown in Table 6.15 and 6.16.

Table 6.15 NPV obtained from CCF rule

Activity	NewTs	t_j	PV
1	7	6	266.08
2	8	7	452.93
3	13	12	11.79
4	1	0	310
5	12	11	429.35
6	15	14	-7.56
7	1	0	-300
8	7	6	-17.74
		NPV	1162.60

NPV = 300exp(-6α)+521exp(-7α) +15exp(-12α) + 310 + 535exp(-11α) +(-10)exp(-14α) + (-300)+(-20)exp(-6α)

NPV obtained from CCF = 1162.60.

Table 6.16 NPV obtained from m-CCF rule

Activity	New Ts	t_j	PV
1	1	0	300
2	9	8	443.97
3	13	12	11.80
4	2	1	303.86
5	8	7	465.11
6	15	14	-7.56
7	1	0	-300
8	7	6	-17.74
		NPV	1199.44

NPV = 300 + 521exp(-8α) + 15exp(-12α) + 310 exp(-α) + 535exp(-7α) + (-10)exp(-14α) + (-300)+ (-20)exp(-6α)

NPV obtained from m-CCF rule = 1199.44 which is increased by CCF 3.1 %.

6.3 Summary

The NPV of the m-CCF heuristics is higher than the frequently used MINSLACK heuristic, which has been found effective in solving the duration minimisation version of this problem.

Table 6.17 summarises the NPV performance of the m-CCF and three heuristic scheduling rules.

Table 6.17 NPV performance of the m-CCF and three heuristic scheduling rules

Rule	MINSLK	m-CCF	GNS	m-CCF	CCF	m-CCF
NPV	121.24	123.12	662.63	832.76	1162.60	1199.44
*Percentage difference (%)	1.52		20.43		3.1	

*Percentage difference = (m-CCF solution-other rule)/ m-CCF solution x 100%

m-CCF can be used to enhance the NPV by adding the time factor to the cumulative cash flow. This new rule should better reflect the time impact of cash receipt and disbursement. The increasing NPV shows that m-CCF provides to superior results to those obtained using the CCF rule. It can be seen that the m-CCF method not only performs excellently, but also is superior to the other three rules.

Chapter 7 Results II: [m-CCF] Validation

7.1 Introduction

This chapter contains the second and third phase experiments that have been conducted to test the effective of new heuristic rule, m-CCF. The NPV performance of the proposed heuristic scheduling rule is analysed. The proposed priority rule based heuristic is implemented in two schemes (serial and parallel) and two strategies (forward and backward). Finally, m-CCF is evaluated by comparing with optimal solution through test problem and validated through three different projects. All the problems in this chapter are solved using the Solver tools of Microsoft office Excel to facilitate searching the optimal solution. More descriptions about the software can be found in Appendix K.

7.2 The Second Phase Experiment

In the second phase, m-CCF method is implemented in two schemes (serial and parallel) and two strategies (forward and backward), and then it is compared with the optimal solution.

The optimal solution presented in this section is based upon the binary integer programming formulation of Doersch and Patterson [162] that have been discussed on Chapter 4. Unfortunately, this technique can only solve unrealistically small problems of marginal practical value. The solution is solved with GRG optimisation routine and used as an optimal solution in this research. All the problems are solved using the Solver tools of Microsoft office Excel to facilitate searching the optimal solution.

7.2.1 Serial and Parallel Scheduling Generation Scheme

This research provides an extensive comparison of the parallel and the serial scheduling scheme. The efficiency and effectiveness of each proposed procedures is proved through comparing the results with the model from Doersch and Patterson [162].

In this research, the optimisation criteria from [162] is solved by using GRG optimisation methods and represents an optimal solution. All the problems are solved using Solver tools of Microsoft office Excel.

The project NPV obtained from the m-CCF methods and the NLP model together with the percentage difference between the m-CCF NPV and the optimal NPV are presented in Table 7.1 and 7.2. Full results can be found on Appendix D.

Table 7.1 NPV obtain from m-CCF method

Output/Solution	m-CCF solution	
	SSGS	PSGS
NPV	700.11	689.41

Table 7.2 NPV obtain from optimal technique

Output/Solution	Optimal solution
NPV	726.11

Table 7.3 Percentage difference between m-CCF and optimal technique

Solution	*Percentage difference (%)
SSGS	3.58
PSGS	5.05

* Percentage difference = (Optimal solution- m-CCF solution)/Optimal solution x 100%.

The project NPV of both serial and parallel scheme are very close to the optimum; the percentage difference of SSGS is only 3.58% and PSGS is 5.05 %. Evidently, the m-CCF performs well in both schemes.

7.2.2 Forward-Backward

The serial and parallel scheduling schemes for constructing feasible schedules are extended by the flexible use of different planning directions (forward and backward).

Generally, all the schemes so far are forward. Within backward, an available task is selected in each iteration and scheduled as late as possible without violating the precedence and resource constraints. Usually, the schedule obtained will not start at the beginning of the planning horizon.

Table 7.4 summarise the NPV performance of the m-CCF heuristic scheduling rules in both scheme. The m-CCF method not only performs excellently, but also is superior to the other three rules that lead to very close to optimal solution. Full results can be found on Appendix E.

Table 7.4 Comparison of NPV from m-CCF and Optimal technique

	m-CCF solution		Optimal solution
	SSGS	PSGS	
Forward	700.11	689.41	726.11
Backward	683.29	649.22	

Table 7.5 Percentage difference between m-CCF and optimal technique

m-CCF solution	*Percentage difference (%)	
	SSGS	PSGS
Forward	3.58	5.05
Backward	5.90	10.59

* Percentage difference = (Optimal solution- m-CCF solution)/Optimal solution x 100%.

Obviously, the project NPV of the forward m-CCF method is closer to the optimum, and the percentage difference is only 3.58%. In both cases, the percentage difference of the forward strategy outperformed that of the backward one. Obviously, the forward method provides the good results when compared with backward.

7.3 The Third Phase Experiment

In this section, the m-CCF rule is test with different data set. These problems consist of projects with between 100 and 300 activities where the project data set are obtained from the PTT Company limited in Thailand in which the author has got accessed to during the Industrial placement periods. All three problems are solved with each of the proposed heuristics procedure described, [m-CCF]-SSPS, -PSGS, -B-SSPS and B--PSGS then compare them with the actual NPV of the project. This is done in order to assess the efficacy of each of the heuristic scheduling rules under different cash flow patterns.

The proposed procedure for solving capital constrained problems is applied and tested by solving on three projects to reduce the capital expenditure. Project 1 represents a general case consists of 102 activities, project 2 with activities consists of 298 activities and project 3 is similar to the project 1 yet contains more critical paths and capital constrained.

The solution for each sample is calculated. The efficiency and effectiveness of each proposed procedures are proved through comparing the results with the actual NPV.

Table 7.6, 7.7 and 7.8 summarises the NPV performance of the m-CCF rule when applying project 1, 2 and 3 respectively. Full results for project 1-3 can be found on Appendix G, H and I.

Table 7.6 NPV performance of the m-CCF rule when applying in project1

Project 1	m-CCF Scheme	
	SSGS	PSGS
Forward	390.67	381.01
Backward	398.67	397.74

Table 7.7 NPV performance of the m-CCF rule when applying in project2

Project 2	m-CCF Scheme	
	SSGS	PSGS
Forward	193.36	190.49
Backward	270.72	213.96

Table 7.8 NPV performance of the [m-CCF] rule when applying in project3

Project 3	m-CCF Scheme	
	SSGS	PSGS
Forward	358.67	361.13
Backward	392.06	394.74

The results of the proposed m-CCF heuristics for all performance measures are given in Table 7.9, and for each solved problem the heuristic rule which got the best results for all performance measures appears in bold.

Table 7.9 Comparison between m-CCF in 3 projects

Project	m-CCF solution				Actual NPV
	Forward		Backward		
	SSGS	PSGS	SSGS	PSGS	
1	390.67	381.01	**398.67**	397.74	388.42
2	193.36	190.49	**270.72**	213.96	203.12
3	358.67	361.13	392.06	**394.74**	388.42

The percentage difference between m-CCF and actual NPV are given in Table 7.10.

Table 7.10 Percentage difference between m-CCF and optimal technique in 3 projects

Project	*Percentage difference (%)			
	Forward		Backward	
	SSGS	PSGS	SSGS	PSGS
1	**+0.58**	-1.91	**+2.64**	**+2.39**
2	-4.80	-6.21	**+33.28**	**+5.34**
3	-7.71	-7.02	**+0.94**	**+1.64**

*Percentage difference = (Actual NPV- m-CCF solution)/Actual NPV x 100%.

According to Table 7.10, for each the heuristic rule which increased the actual NPV appears in bold, the proposed m-CCF heuristic method successfully improved the NPV, and according to the final results, it is obvious that the backward strategy has a better situation than the forward scheme. The summary of results is depicted in Table 7.11.

Table 7.11 Summary of results of comparing the proposed techniques

Methods	Three Test problems	
	Number of best solutions obtained (NPV)	Percentage of best solutions obtained
[m-CCF]-SSGS	0	0
[m-CCF]-PSGS	0	0
[m-CCF]-B-SSGS	2	66.66
[m-CCF]-B-PSGS	1	33.33

7.4 Summary

The Second phase

The project NPV of SSGS is closer to the optimum than PSGS; the percentage difference of SSGS is only 3.58% and PSGS is 5.05 %.

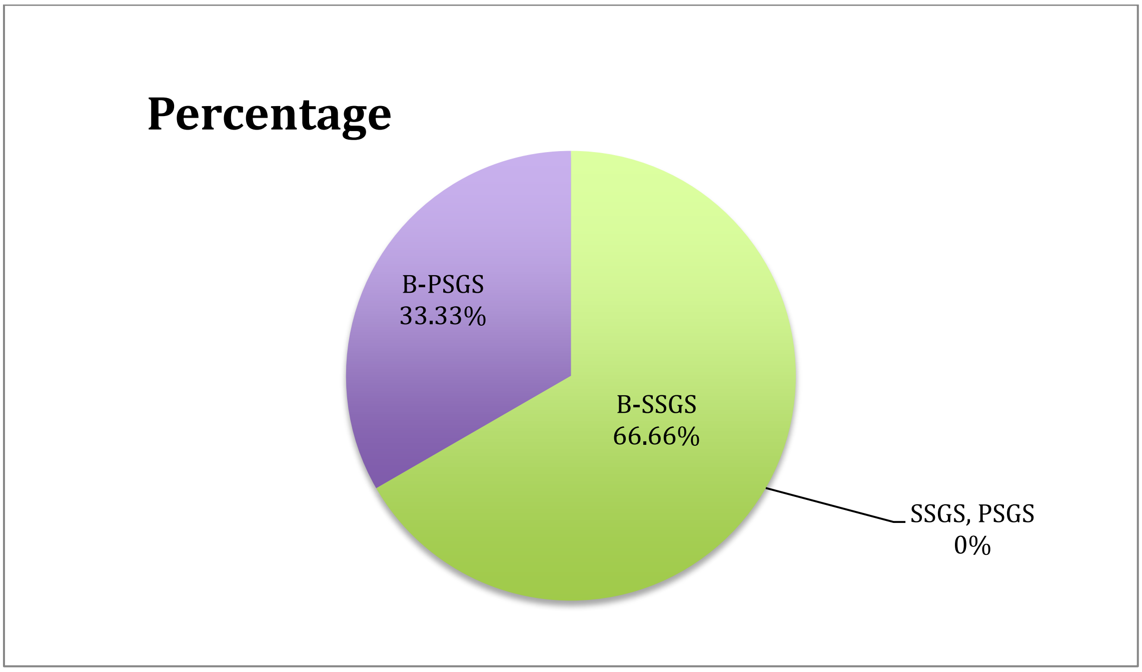
The project NPV of the forward m-CCF method is closer to the optimum, and the percentage difference is only 3.58%. In both cases, the percentage difference of the forward strategy outperformed that of the backward one. Obviously, the forward method produces the good results when compared with backward.

The Third phase

In most cases, SSGS outperformed PSGS except for the third project. However, the serial method does not generally perform better than the parallel method. The complexity of project also affects the performance of the chosen technique and NPV. In contrast to the second phase, it can obviously be observed that a combination of a scheduling scheme and a priority rule may yield a good result applied in backward direction and a bad result in forward direction during the third phase of experiment.

Figure 7.1 indicates the percentage of first place obtained by each method for the number of best solution compared to other methods in all three solved test problems in this study.

Figure 7.1 The percentage of each method obtaining best solution



In accordance with this pie chart, whereas the proposed [m-CCF]-SSGS rule obtained the best result in 66.66% of problems (2 test problems out of 3) compared to the other methods, it is evident that this method takes first place among the other methods. In addition, B-PSGS, which achieves the best results in only 1 problem out of 3 (33.33%), is in second place. Moreover, F-SSGS and F-PSGS are in third place for not achieving the best results of the problems solved.

Chapter 8: Discussion and Recommendations

8.1 Introduction

This chapter provides a review of each research question, which has been mentioned earlier in this thesis. In the end, research findings are discussed and recommendations are made.

8.2 Research Finding and discussion

This research has developed a heuristic technique for the constrained project scheduling problem. Doersch and Patterson [162] defined the capital-constrained project scheduling problem (CCPSP) as one of scheduling a project that take place over the course of the project, where investment in project activities is constrained by a capital constraint. Typically project investors will place a constraint on the amount of funds that can be outstanding on work on project activities at any point in time.

The capital constraint is usually imposed on a project to limit the amount of capital that may be expended per period for internal resources, suppliers, joint venture partners, and subcontractors for project activities. In organisations with limited capital to invest in new and continuing projects, reinvesting progress payments provides for internal accountability for completing portions of the projects and a source of capital in addition to that provided by investors and partners that can be used for earlier scheduling of project activities. Thus, the amount of capital available in each period is a renewable resource, where the investors make the same amount of capital available in each subsequent period of the project, plus any progress payments (cash inflows) that are reinvested in the project.

This section discusses the research findings according to each stage of the experiment as following;

8.2.1 The First phase experiment

In the first phase, the proposed m-CCF heuristic solution is compared with other rules which are MINSLK, GNS and CCF. Only the small test problems are included in this phase, since solving a very large-scale problem would require a very large amount of computational effort. This comparison is made by using Network Diagram adapted from literature.

Summary results are given in both Chapters 6 and 7. The NPV of the m-CCF heuristics is consistently higher than that of the frequently used MINSLACK heuristic, which has been found effective in solving the duration minimisation version of this problem.

However, the MINSLACK heuristic does perform well in those instances in which a high final (positive cash payment is present) [165]. As the positive cash inflows occur toward the end of a project, the heuristics that are effective in minimising project duration will be resulting in high NPV solutions as well. This is because, depending on the concentration of positive and negative cash flow activities, the MINSLACK rule potentially ignores cash flows associated with intermediate project activities. Naturally, when all or most of the positive cash flow activities are found at or near the end of a project, procedure which yield shorter duration schedules are also likely to produce schedules resulting in high net present value amounts. Furthermore, the m-CCF method outperforms all of the other rules, which are GNS and CCF decision rules. CCF considers only the cash flows and GNS considers only the number of all follower activities. m-CCF combined them both and adds time value factors to the cash flow. It is found that the m-CCF results in higher NPVs than any of other heuristics.

8.2.2 The Second phase experiment

This research has revisited schedule schemes, which can be applied in heuristic approaches for solving the capital-constrained project scheduling problem. Two of the oldest and the best known heuristics are the serial and the parallel scheduling scheme, respectively.

The majority of publications dealing with scheduling schemes for the PSP report on the performance of one scheme when applied as a single-pass approach only [189]. This research is provided an extensive comparison of the parallel and the serial scheduling scheme. The m-CCF method is embedded in two schemes (serial and parallel). Both methods generate feasible schedules, which are optimal in the absence of resource restrictions. More detail on both schemes will be discussed on the next phase.

The serial and parallel scheduling schemes for constructing feasible schedules are extended by the flexible use of different planning directions (forward and backward).

The m-CCF method is also implemented in two strategies (forward and backward). First of all, the experiments reveal that the planning direction has a considerable influence on the performance of priority rule-based heuristics and, hence, the scheduling scheme employed. A combination of a scheduling scheme and a priority rule may yield a good result applied in forward direction and a bad result in backward direction. This may be just the other way round from the work done by Smith-Daniels and Aquilano [38] and in the third phase of this research. This is because, depending on the concentration of positive and negative cash flow activities, delay the negative cash flow as much as possible possibly results in reducing cost and maximising NPV. However, it can be observed that the forward method generates active schedules while the backward scheduling scheme creates non-delay schedules.

8.2.3 The Third phase experiment

In the third phase, the main purpose is to address an area of strengths and weakness in the chosen technique. m-CCF is tested in three different dataset.

All three problems are solved with each of the proposed heuristics procedure described, [m-CCF]-SSPS, -PSGS, -B-SSPS and -B-PSGS. Then, it is compared with the actual NPV of each project. This was done in order to assess the efficacy of each of the heuristic scheduling rules under different cash flow patterns. It can be seen that the greater the number of activities, the more complex the scheduling problem becomes, and the greater the impact upon the performance of the scheduling rule.

The [m-CCF]-B-SSGS method has been shown to produce the best solution on most occasions, compared with the actual NPV. In most cases, SSGS outperformed PSGS except for the third project. That means that the complexity of project also affects the performance of the chosen technique and NPV. From the results, it can be observed that the parallel method does not generally perform better than the serial method. The parallel schedule scheme searches in a smaller solution space than the serial schedule scheme but with the drawback that when considering a regular performance measure - the solution space might not contain an optimal schedule. Hence, the serial method is superior for large sample sizes and, for instances, which are only moderately resource-constrained. This insight should be of importance when deriving fast problem-specific parameter-guided heuristics.

The experimental results demonstrate clearly that the backward m-CCF method is a superior than the forward m-CCF heuristic scheduling rule in reducing capex. Even though the implementation of the forward- and backward does not guarantee a good result, it is vital to develop a good initial solution rule and designing excellent improvement iterations to increase the total project NPV. It is recommended that bidirectional planning should also be included in other heuristics using scheduling schemes such as sampling procedures and meta-heuristic-based approaches. By combining both scheduling schemes with a subset of successful priority rules and employing unidirectional and bidirectional planning, a very efficient multi-pass heuristics can be designed.

8.3 A Review of the research questions

(i) Is NPV the most effective measure for evaluating project investment?

The work done by Flaig [2] also concludes that NPV is by far a better method to evaluate capital investments as compared to the other methods. The main reason NPV is preferable is the fact that one can always calculate NPV given the discount rate. Compare this to the IRR which may not always exist or it may not be unique. NPV provides an investor with a money amount that an organisation tends to gain or lose by investing in a project. Other measures such as payback period provide time periods as compared to money amounts. The main advantage NPV has over IRR is that one is always able to calculate it with a discount rate whereas IRR may not yield a value in certain instances or there may be multiple IRR values.

From the authors opinion, NPV is important to an organisation that is about to undertake a capital budgeting project since the organisation will be able to judge how much of its capital investment will make a return on investment. NPV may be the most appropriate methods, when comparing mutually exclusive projects or when budget rationing is the option due to scarce resources, which the organisation has at its disposal. IRR method may not result in a solution in certain cases or it may have multiple solutions. IRR is a rate of interest yet NPV provides a money amount that is either made or lost if an organisation were to invest in a capital project.

(ii) Can the existing priority rule based heuristics scheduling techniques be improved?

The review of the literature reveals that the development during the last years, priority rule-based methods has attracted more attention than meta-heuristic approaches again. Klein [189] confirmed that priority rule-based heuristics are in wide and general use due to yielding acceptable results with a reasonable computational effort. It is a very good idea to employ more effective scheduling schemes within such procedures. Even though recombining merely existing ideas occasionally seems to be less creative than developing new ideas, some of the integration efforts have put well-known techniques into a new and promising context, and the results have often been encouraging [168].

This research proposes the new rule that combines the simplicity of the priority rule heuristic scheduling but at the same time adding more components. Many methods consider both scheduling directions instead of only forward scheduling, more than one type of local search operator, or even more than one type strategy.

In the author's own research the evidence suggests that the m-CCF method not only performs excellently, but also is superior to the other three rules that lead to very close to optimal solution. In Chapter 6 and 7, the proposed method indicates the NPV performance of the m-CCF in both schemes. The proposed m-CCF heuristic method successfully improved the NPV of the project.

(iii) Is it possible to optimise the NPV of the project subject to a late start scheme?

The review of the literature has been compared the duration and NPV of a late-start critical path schedule to that of an early-start critical path schedule. Smith-Daniels and Aquilano [38] assumptions were tested using the 110 example problems from Patterson [147]. An improved average NPV and lower average duration can be found for late-start schedules than early-start schedules. They concluded that a heuristically determined right shifted schedule yields a higher NPV and lower average duration than schedules derived with heuristics that schedule each activity as early as possible.

The research by Ulusoy and Özdamar [116] present an iterative scheduling algorithm with the objective of improving both the project duration and NPV. The consecutive forward/backward scheduling passes made by the iterative algorithm result in a smoother resource profile, along with right shifting of activities, improves both the project duration and NPV. In the cash flow model assumed here, activity expenditures occur at their starting times and payment is made on completion of the project. The algorithm was tested on two sets of problems from the literature. The results demonstrated that under the assumed cash flow model, the iterative scheduling algorithm improved both criteria.

In the author's own research, the outputs reveal that for some instance, a combination of a scheduling scheme and a priority rule may yield a good result applied in forward direction and a bad result in backward direction. For another instance, this may be just the other way round. According to the final results, it has been clearly found that the backward strategy has a better situation than the forward in reducing cost.

In the author's opinion, the bi-directional generation scheme should be used for more advanced problem formulations. The scheduling schemes can be extended to assign starting times bi-directionally, i.e. to construct schedules in forward and backward direction simultaneously. The serial and parallel scheduling schemes for constructing feasible schedules are needed to be extending by the flexible use of different planning directions including a bidirectional planning. By combining both scheduling schemes with a subset of successful priority rules and employing unidirectional and bidirectional planning, a very efficient multi-pass heuristics can be designed.

(iv) Does project complexity affect the efficiency of the priority rule-based scheduling techniques?

The review of the literature suggests that the greater the number of projects combined, the more complex the scheduling problem becomes, and the greater the impact upon the performance of the scheduling rule. The research by Chiu and Tsai [192] confirms this by comparing the total project NPV performance among the five rules and also varied significantly with any increase in the number of projects combined.

Valls et al. [193] reported the comparison of the serial and the parallel scheduling that none of the schemes is dominant. The assumption made by A1varez-Valdes and Tamarit [119] that parallel algorithms *"seem to work better than the serial ones"*.

In the author's own research, the outputs reveal that project complexity indeed affect the efficiency of the priority rule based scheduling techniques. In Chapter 6 and 7, the proposed method indicates that the greater the number of projects that were combined, the more complex the scheduling problem became, and the greater the impact upon the performance of the scheduling rule.

From the results, it can be observed that the serial method is superior for large sample sizes. The parallel scheduling scheme suits for a smaller problem and might not contain an optimal schedule. However, the parallel seems to be more accurate methods as it searches in a smaller solution space than the serial scheduling scheme. In conclude, the serial is recommended for large and complex problems while the parallel can be served when the project scheduling problem is small.

8.4 Recommendations

This method is to improve an existing solution that managers often follow in practical field. It is suggested that the proposed technique can be implemented in two ways.

I. If a project manager heavily emphasises the accuracy of project scheduling and the project scheduling problem is small (in general, the number of total activities is less than 50), then the m-CCF parallel scheme is recommended. It can be served as a benchmark for evaluating the performance of other heuristic rules developed in the future.

II. If a project manager primarily concerns with scheduling efficiency and effectiveness, or the project scheduling problem is large and complex, the m-CCF serial scheme is recommend.

[m-CCF]-SSGS method has several advantages, and can be implemented effectively in the complicated real-world situation. In this study, solutions obtained are found to be efficient solutions. Thus, a simple approach may be more suitable for applying to real life projects that consist of large number of activities. Furthermore, it can easily be extended to solve other types of time-cost trade-off problems, when more constraints are added or more factors are considered.

Its notion is very simple and the method is easy to apply. Therefore, this rule can be applied to the RCPSP (resource-constrained project scheduling problem) with an objective function that maximises the total project NPV or minimises the total project delay. Finally, the m-CCF priority rule-based scheduling method can assist project managers in effectively planning and scheduling their company's limited resources. As a result, the NPV of the project will be further improved.

However, a need for more robust and practically meaningful tool/framework for the assessment of project NPV is required, as recommended by the practitioners also. The m-CCF algorithm should be tested on additional real-world projects so as to determine its applicability in a variety of industries.

The solution generated for the engineering project discussed in the previous section was acceptable to the engineering personnel who would use it and they felt that it would be useful for project control. They felt that it would improve their utilisation of resources and enable more efficient re-scheduling. Particularly in the construction industry, this would provide the contractor with a method for improving estimating performance and control of material and labour usage. It would also allow the contractor to integrate the costing function with the project schedule so that progress payments could be easily supported and justified.

However, the results in each of the example problems are dependent on a variety of environment variables. These factors include capital costs and various project structure characteristics. These should be considered in the testing of heuristic solution methods, since the effectiveness of a heuristic will most likely to be correlated with the levels of the various factors.

Finally, the approach considered in this research should be extended to multiproject environments, where it is necessary to make capital allocation decisions between competing projects. The project manager would then receive assistance in one of the most difficult decisions in project management.

Research in each of these areas will improve the viability of this approach to the project scheduling problem, but the development of heuristics should, in the short term, provide the most assistance to project managers in managing large projects. The consideration of capital costs and constraints as well as institution of a monetary objective function should serve to improve project return on investment.

Chapter 9: Conclusions and Future work

9.1 Introduction

This final chapter provides a conclusion of the research. A contribution to knowledge and its implications to academic and industrial perspective are highlighted. In the end of this chapter, recommendations are made for future research.

9.2 Conclusions

In order to collectively satisfy the research aim as mentioned above, a number of research objectives were developed. This section reviews and highlights the extent to which those objectives are accomplished through the various phases of the research.

Objective 1: To investigate the approaches which have been developed for solving the project scheduling problems and cash flow management.

The literature review revealed that project scheduling problem was solved by two distinctly different approaches. The first approach includes mathematical techniques that produce optimal solutions. Several optimisation models were developed like those given by Russell [158] Grinold [159] Elmaghraby and Herroelen [161] Doersch and Patterson [162]. Although mathematical techniques produce optimal solutions, they fail to solve the relatively medium-size and more complicated problems usually encountered in practice. The second approach includes heuristic methods. They are based on a process of decision making according to a set of priority rules that are based on activity characteristics. Different models are formulated using different combinations of priority rules.

In this research, several heuristics have been described as a simple approach to implement (especially when compared to alternate approaches for solving the Max NPV version of this problem), and possess built-in, forward-looking mechanisms through which improved schedules result.

Objective 2: To examine the significant relationship between each capital budgeting technique.

The review of the literature reveals that NPV and IRR are two popular methods used by organisations in evaluating investments that require capital budgets. These two techniques are used in conjunction, however, each of these methods has its pros and cons. Both methods are widely used and in some instances NPV is seen as a better and superior measure for the reason that in some cases there may not exist an IRR or if does it may not be unique.

NPV and IRR conflict may arise when the timings of cash flows are not at par with each other whilst comparing more than one project. The research by Lurin [95] discusses this conflicting results for IRR and NPV where one project seems to have a lower IRR and higher NPV whereas another project has a higher IRR and lower NPV. In such a conflict making a decision based on NPV is more reliable as compared to making one based on the IRR. The right way to rank projects is to use the NPV that they generate and their required peak funding, not the IRR.

In conclusion, it demonstrates that no single technique can paint a complete picture of the attractiveness of the project, and therefore a combination of these techniques is normally used to make an investment decision. Which technique is of prime importance depends on the situation of the investor. However, with no limitations, NPV would probably be the primary method.

Objective 3: To evaluate the performance of existing scheduling rules and techniques.

In the first phase, the proposed m-CCF heuristic solution is compared with other rules which are MINSLK, GNS and CCF. Only the small test problems are included in this phase, since solving a very large-scale problem would require a very large amount of computational effort. This comparison is made by using Network Diagram adapted from literature.

Summary results are given in both Chapters 6 and 7. The NPV of the m-CCF heuristics is consistently higher than that of the frequently used MINSLACK heuristic, which has been found effective in solving the duration minimisation version of this problem.

However, the MINSLACK heuristic does perform well in those instances in which a high final (positive cash payment is present) [165]. As the positive cash inflows occur toward the end of a project, the heuristics that are effective in minimising project duration will be resulting in high NPV solutions as well. This is because, depending on the concentration of positive and negative cash flow activities, the MINSLACK rule potentially ignores cash flows associated with intermediate project activities. Naturally, when all or most of the positive cash flow activities are found at or near the end of a project, procedure which yield shorter duration schedules are also likely to produce schedules resulting in high net present value amounts. Furthermore, the m-CCF method outperforms all of the other rules, which are GNS and CCF decision rules. CCF considers only the cash flows and GNS considers only the number of all follower activities. m-CCF combined them both and adds time value factors to the cash flow. It is found that the m-CCF results in higher NPVs than any of other heuristics.

Objective 4: To consider an alternative heuristic scheduling technique with improves performance.

This research presents m-CCF scheduling heuristic that is effective at maximising NPV of capital-constrained project. The m-CCF priority rule technique operates by dynamically selecting an activity with highest m-CCF value from a list of available activities without violating precedence, critical path and other constraints. m-CCF is the modified version of CCF. However, rather than considering only the cash flows or the number of all follower activities, m-CCF use the discounted value of all future cash flows of successor activities. Discounting activity cash flows should better reflect the time impact of cash receipts and disbursements. Thus, this technique adds some discount factors to the undiscounted cash flow.

The m-CCF method is embedded in two schemes (serial and parallel). Both methods generate feasible schedules, which are optimal in the absence of resource restrictions. It was proven that the serial method generates active schedules while the parallel scheduling scheme creates non-delay schedules. Hence, the parallel scheduling scheme searches in a smaller solution space than the serial scheduling scheme but with the drawback that when considering a regular performance measure - the solution space might not contain an optimal schedule.

From the results, it can be observed that the parallel method does not generally perform better than the serial method. Rather, it provides only good results for single-pass scheduling and small sample sizes as well as for "hard" (that is highly resource-constrained) problems. Hence, the serial method is superior for large sample sizes and for instances which are only moderately resource-constrained. This insight should be of importance when deriving fast problem-specific parameter-guided heuristics.

The constrained project scheduling problem belongs to the class of NP-hard problems, and hence, many heuristic solution procedures have been developed and described in the literature. Many research papers, however, focus on the development of single-pass algorithms in which activities are ranked by a priority vector determining the order of resource allocation during a schedule generation process. Since these methods can only generate a single solution, this study is extended by improvement methods and/or backward scheduling schemes.

Objective 5: To apply the model to a wide range of different projects.

This study examines the schemes by implementing the m-CCF heuristic selection methods. In the second and third phase results section, the output for both serial and parallel schemes is compared with a forward and backward generation scheme, along with the optimal solution method.

The m-CCF rule is test with different data set. These problems consist of projects with between 100 and 300 activities where the project data set are obtained from the PTT

Company limited in Thailand in which the author has got accessed to during the Industrial placement periods. All three problems are solved with each of the proposed heuristics procedure described, [m-CCF]-SSPS, -PSGS, -B-SSPS and B--PSGS then compare them with the actual NPV of this projects. This is done in order to assess the efficacy of each of the heuristic scheduling rules under different cash flow patterns.

Test results demonstrate that both serial and parallel generation scheme are able to produce optimal results. Moreover, the results show that in some instances the use of the forward improves the results dramatically, for another instance; this may be just the other way round. The use of bidirectional planning is recommended for the further research.

According to the results presented, it appears to be worth-while to employ heuristics based on serial m-CCF in general, and to employ parallel scheme m-CCF methods under a more restrictive set of assumptions. At the expense of a modest increase in computation time, it is beneficial to use forward-backward solution, for schedule improvement, selecting the schedule with the largest net present value among those available.

Objective 6: Propose the new improve heuristic scheduling technique to minimise the capital expenditure.

The propose m-CCF scheduling heuristic appear to be effective at maximising NPV of capital-constrained project. The m-CCF method is also implemented in two strategies (forward and backward). First of all, the experiments reveal that the planning direction has a considerable influence on the performance of priority rule-based heuristics and, hence, the scheduling scheme employed. A combination of a scheduling scheme and a priority rule may yield a good result applied in forward direction and a bad result in backward direction. This may be just the other way round from the work done by Smith-Daniels and Aquilano [38]. This is because, depending on the concentration of positive and negative cash flow activities, delay the negative cash flow as much as possible possibly results in reducing cost and maximising NPV.

In the third phase experiment, all three problems are solved with each of the proposed heuristics procedure described, [m-CCF]-SSPS, -PSGS, -B-SSPS and -B-PSGS. Then, it is compared with the actual NPV of each project. The results demonstrate clearly that the backward m-CCF method is a superior than the forward m-CCF heuristic scheduling rule in reducing capex.

<u>The m-CCF heuristic model Validation</u>

This research seeks to provide an in-depth investigation of the project scheduling methods with the aim of proposing improvements to the existing method. Author proposes a new heuristic technique with an embedded priority rules to optimise the NPV of cash flows for projects.

The research has started by reviewing the current scheduling techniques and addressing the problems. Finally, the m-CCF has been proposed to optimise the NPV of cash flows for projects. Validation of this method will be accessed through comments from practitioners. On April/May 2011, the author spent between two months duration of the research working as an intern in Gas business group at PTT Company limited, the huge oil& natural gas Company in Thailand. This provides the opportunity to collect primary data and receive some comments and feedbacks from there.

The comments are list below;

"The results clearly indicate that the use of m-CCF heuristics in project management is very promising. Literally, anything that cannot be easily solved by conventional exact optimisation techniques. Most existing commercial project management software packages only provide limited project scheduling, project tracking and reporting aids and fall short in their computational capabilities. It is our belief that m-CCF heuristics offer a rich set of computational techniques that can greatly enhance current project management tools."

Mr Sompong

The PTT Gas Pipeline Project director

"Despite of some deficiencies, it is our belief that the potential of m-CCF heuristics is high. However, efforts need to be made to show that m-CCF heuristics can be useful to solve complicated real world problems. To bridge the gap between a real world problem and a formulated model, the model must be formulated based on reasonable assumptions.

The power of m-CCF heuristics has not been utilised in commercial project management software packages today, but the picture could be different in the future. It is in the best interest of major software developers or project management firms to work with researchers to further validate the potential benefits that m-CCF heuristics can bring to real world project management."

Mr. Surachart

Senior Planning Analyst

"The outcomes from this research suggest that the m-CCF technique is achievable. General deficiencies of current technique are highlighted. This could add considerable value to our business and become standard practice in our common process. However, some topics required for further studies to identify with the hope that researchers, both new comers and experienced veterans, would pick up some of these ideas and work on them to further advance this area of research."

Mr Vasin

Senior Project Manager

Considering all the positive comments and feedbacks, the results of using m-CCF heuristic method is high. This method could become standard practice due to yielding acceptable results with a reasonable computational effort and can be applies in realistic problems. It also has a huge practical significance and could add considerable value the real world's business.

9.3 Contribution to Knowledge

This research has implications for both practitioners and academicians. The outcomes of this research contain originality and provide an addition to the academic body of knowledge and practical implications in the area of project scheduling in particular.

Since the development of critical path methods (CPM), there have been many different scheduling models based on this technique. Some of these models are mathematical (analytical) algorithms, which aimed to produce optimal scheduling solutions. However, their application proved to be successful for research and academic purposes but not for practical and complicated scheduling problems encountered in construction companies [124]. Because of this lack of success with the optimisation procedures, major efforts have been expended in developing heuristic scheduling procedures, in which the main objective is to produce feasible and good solutions.

For real-world problems with a large number of jobs, heuristics such as priority rule-based procedures are among the methods of choice to schedule. These heuristic scheduling procedures depend on assigning priorities to scheduling activities using heuristic rules. The performance of these rules under different conditions, and using different approaches has been tested and reported in many published papers.

This research contributes to existing knowledge by:

- This research has developed a heuristic technique for the capital constrained project scheduling problem. The algorithm proposed successfully combines priority rules methods with backward–forward scheduling. In addition, the algorithm includes as a determinant characteristic the alternative use of the serial and parallel schedule generation schemes in such a way that it benefits from the properties provided for both of them.

- The results of the computational experience indicate that the propose rules is able to outperform the best currently available methods, regardless of the project size. In fact, when the number of activities of the project increases, the new technique increases its effectiveness, maximising the project NPV even more with respect to the scheduling methods compared.

- The proposed algorithm is easy to code and it is very fast due to the fact that it uses activity lists. These characteristics and its great performance favour the idea of its adaptation to solve more general project scheduling problems such as muti-mode and multi-project as well as the consideration of other performance measures in addition to time.

- These approaches are managerially significant because they are simple to compute in the context of project management and intuitively based on scheduling theory. Thus, it requires less computational effort than optimisation-based approach.

- In conclusion, according to the provided results, the combination of priority rule methods and backward–forward scheduling can be considered as a good direction to develop further heuristics that can be built as a powerful tool in project planning and control systems. In fact, the proposed heuristic can easily be integrated into commercial project management software such as Microsoft Project, CA-Super Project or Time Line using the programming language included, thus improving their capabilities.

9.4 Future works

Future intentions are as follows:

- The need to develop more advanced meta-heuristic search procedures to extend the basic problem type, for example, multi-mode scheduling problems, pre-emptive activity execution, variable cash flows and many more.

- The author believes that the bi-directional generation scheme and the recursive forward/backward improvement method can still be used for more advanced problem formulations. The scheduling schemes can be extended to assign starting times bidirectionally, i.e. to construct schedules in forward and backward direction simultaneously. The serial and parallel scheduling schemes for constructing feasible schedules are needed to be extending by the flexible use of different planning directions including a bidirectional planning.
- A possible extension of the analysis is to consider the case of stochastic activity times. This leads to the problem of trade-off between the present cost of a project and the probability of completing it on schedule. Such analysis under the assumption of stochastic activity times should be the type of analysis offered by the next generation of commercial project management software.
- The variation of different type of constraints is limited. In order to validate and establish the factors for project; it is recommended that similar research may be done in more different type of constraints. This will not only help to validate contextual aspect but would also help to investigate any possible variations.
- This method was tested with number of project networks. However the variation of complexity of project was limited. It is recommended that further research can be done by exploring more on different type of projects.
- The consideration of develop a new emerging theory of investment on max NPV rule under uncertainty can possibly shed some new light on the complex field of project scheduling.

References

1. Hastings, N.A.J., *On resource allocation in project networks.* Operational Research Quarterly 1972. **23**: p. 217-221.
2. Flaig, J.J., *Improving Project Selection Using Expected Net Present Value Analysis.* Quality Engineering, 2005. **17**: p. 535–538.
3. Bagajewicz, M., *On the Use of Net Present Value in Investment Capacity Planning Models.* Ind. Eng. Chem. Res, 2008: p. 9413–9416.
4. Davis, E.W., *Project Scheduling under Resource Constraints Historical Review and Categorization of Procedures.* IIE Transactions, 1973. **5**: p. 297 -313.
5. Willis, R.J., *Critical Path Analysis and Resource-Constrained Project Scheduling—Theory and Practice.* European Journal of Operational Research, 1985. **21**: p. 149–155.
6. Clough, R.H., G.A. Sears, and S.S. K., *Construction project management*. 4th ed. 2000, New York: John Wiley & Sons.
7. Dayanand, N. and R. Padman, *Payments in projects: a contractor's model.*, 1993, The Heinz School, Carnegie Mellon University, Pittsburgh.
8. Demeulemeester, E. and W. Herroelen, *A branch-and-bound procedure for the multiple resource-constrained project scheduling problem.* Manage Sci, 1992. **38**: p. 1803–1818.
9. Davis, E.W. and J.H. Patterson, *A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling.* Management Science, 1975. **21**: p. 944–955.
10. Patterson, J.H., *A Comparison of Exact Approaches for Solving the Multiple Resource Constrained Project Scheduling Problem.* Management Science, 1984. **30**: p. 854–867.
11. Bey, R.B., R.H. Doersch, and J.H. Patterson, *The net present value criterion: its impact on project scheduling.* Project Management Quarterly 1981. **12**(2): p. 35–45.
12. Elazouni, A.M. and F.G. Metwally, *Finance-Based Scheduling: Tool to Maximize Project Profit Using Improved Genetic Algorithms.* Journal of construction engineering and management© ASCE 2005. **131**(4).
13. Flanagan, R.a.N., G., *Cycle Costing /or Construction". Quantity Surveyors Division of the Royal Institution of Chartered Surveyors*. 1983, London.
14. Drexl, A. and J. Grünewald, *Non preemptive multi-mode resource-constrained project scheduling.* . IIE Transactions 1993. **25**(5): p. 74–81.
15. E Steinberg, E., W.B. Lee, and B.M. Khumawala, *A Requirements Planning System for the Space Shuttle Operations.* Journal of Operations Management, 1980. **1**(2).
16. Gonguet, L., *Comparison of three heuristic procedures for allocating resources and producing schedule*, in *ProJect Planning by Network Analysts*. 1969: North-Holland, Amsterdam.
17. Kapur, K.C. and L.R. Lamberson, *Reliability in Engineering Design*. 1977, New York: John Wiley & Sons.
18. Jahn, F., M. Cook, and M. Graham, *Hydrocarbon Exploration and Production Developments in Petroleum Science*. Vol. 55. 2008: Elsevier.
19. Feibus, A., *Project leaders*. 1998, NewYork: Manhasset.
20. Gormley, F.M. and N. Meade, *The utility of cash flow forecasts in the management of corporate cash balances.* European Journal of Operational Research, 2007. **182**(2): p. 923–935.
21. Harvey, G., *Life-cycle costing: a review of the technique Management Accounting.* 1976: p. 343-347.
22. Lock, D., *Project management*. 7th ed. 2000: Gower, Burlington, VT.
23. Needles, B.E., M. Powers, and H.R. Anderson, *Principles of Accounting*. 7th Ed ed. 1999, Boston: Houghton Miffilin.
24. Au, T. and C. Hendrikson, *Profit measures for construction projects.* Journal of Construction Engineering and Management ASCE, 1986. **112**(2): p. 273–286.
25. Pries, K.H., *Six Sigma for the New Millennium: A CSSBB Guidebook*. 2009: ASQ Quality Press.

26. Carruthers, J.A. and A. Battersby, *Advances in critical path methods.* . Operation Res Quart 1966. **17**: p. 359–80.
27. Conway, R.W., W.L. Maxwell, and L.W. Miller, *Theory of Scheduling*. 1967, NewYork: Addison-Wesley.
28. Radermacher, F.J., *Scheduling of project networks.* Annals of Operations Research 1985/86. **4**: p. 227-252.
29. Robert, K., *Project scheduling with time-varying resource constraints.* International Journal of Production Research, 2000. **38**: p. 3937 – 3952.
30. Apt, K.R., *Principles of constraint programming*. 1st Ed ed. 2003, Cambridge: Cambridge University Press.
31. Barbosa, P.S.F. and P.R. Pimentel, *A linear programming model for cash flow management in the Brazilian construction industry.* Construction Management and Economics Routledge 2001. **19**(5): p. 469–479.
32. Herroelen, W.S., P.V. Dommelen, and E.L. Demeulemeester, *Project network models with discounted cash flows a guided tour through recent developments.* European Journal of Operational Research, 1997. **100**: p. 97-121.
33. Sepil, C. and N. Ortac, *Performance of the heuristic procedures for constrained projects with progress payments.* Journal of the Operational Research Society, 1997. **48**(11): p. 1123–30.
34. Tzafestas, S. and A. Triantafyllakis, *Deterministic scheduling in computing and manufacturing systems: a survey of models and algorithms.* Mathematics and Computers in Simulation, 1993. **35**: p. 397-434.
35. Storer, R.H., S.D. Wu, and V. R., *New search spaces for sequencing problems with application to job shop scheduling.* Management Science 1992. **38**: p. 1495-1509.
36. Moder, J., *Network techniques in project management*, in *Project Management Handbook*. 1988, Van Nostrand Reinhold: New York. p. 324-373.
37. Shaffer, L.R., J.B. Ritter, and W.L. Meyer, *The critical-path method.* . 1965, New York: McGraw-Hill.
38. Smith-Daniels, D.E. and N.J. Aquilano, *Using a late-start resource-constrained project schedule to improve project net present value.* . Decision Sciences 1987. **18**: p. 617–30.
39. Vollman, T.E., W.L. Berry, and C.D. Whybark, *Manufacturing Planning and Control Systems*. Vol. 111. 1984, Irwin: Homewood.
40. Battersby, A., *Network analysis for planning and scheduling*. 1967, New York: St Martin's Press
41. Van Dorp, J.R. and M.R. Duffey, *Modelling Statistical Dependence in Risk Analysis for Project Networks.* International Journal of Production Economics, 1999. **58**: p. 17-29.
42. J., H.T., *Software solutions.* Building design & construction, 1999. **November**: p. 38-40.
43. Liberatore, M.J., B. Pollack-Johnson, and C.A. Smith, *Project management in construction: software use and research directions.* Journal of construction engineering and management, 2001. **ASCE, 127**(2): p. 101-107.
44. Miller, R.W., *How to plan and control with PERT.* Harvard Business Review, 1962: p. 93-104.
45. Malcolm, D.G., et al., *Application of a technique for research and development program evaluation.* Operations Research, 1959. **7**(5): p. 646-669.
46. Bennington, G.E. and C.F. Mcginnis, *A Schedule Modifying Algorithm for Project Planning with Resource Constraints.* NSCO-IE Technical Report, 1972. **73**.
47. Goldratt, E.M., *Critical Chain*. 1997, Great Barrington: The North River Press Publishing Corporation.
48. Raz, T., R. Barnes, and D. Dvir, *A critical look at Critical Chain Project Management.* Project Management Journal, 2003. **34**: p. 24-32.
49. Pinto, J.K., *Some constraints on the theory of constraints: Taking a critical look at the Critical Chain.* PM Network, 1999. **13**(8): p. 49-51.
50. Herroelen, W. and R. Leus, *On the merits and pitfalls of critical chain scheduling.* Journal of Operations Management, 2001. **19**: p. 559-577.

51. Trietsch, D., *Why Critical Path by any other name would smell less sweet?* Project Management Journal, 2005. **36**: p. 27-36.
52. Van Slyke, R.M., *Monte Carlo methods and the PERT problem.* Operations Research, 1963. **11**(5): p. 839-860.
53. Fishman, G.S., *A Monte Carlo sampling plan for estimating network reliability.* Operations Research, 1986. **34**(4): p. 581-594.
54. Cook, M.S. *Real-world Monte Carlo Analysis*. in *Proceeding of PMI Annual Seminars and Symposium*. 2001. Nashville.
55. Yang, K.K., L.C. Tay, and C.C. Sum, *A comparison of stochastic scheduling rules for maximizing project net present value.* European Journal of Operational Research, 1995. **85**: p. 327-339.
56. Bowman, E.H., *The schedule-sequencing problem.* Operations Research 1959. **7**: p. 621-624.
57. Boctor, F.F., *Resource-constrained project scheduling simulated annealing.* International Journal of Production Research, 1996. **34**(8): p. 2335–54.
58. Williams, J.R., et al., *Financial & Managerial Accounting* 2008.
59. Zwikael, O., K. Shimizu, and S. Globerson, *Cultural differences in project management processes: a field study.* Int J Project Manage 2005. **23**(6): p. 454–62.
60. Christofides, N., R. Alvarez-Valdes, and J.M. Tamarit, *Project scheduling with resource constraints: A branch and bound approach.* European Journal of Operational Research 1987. **29**: p. 262-273.
61. Davis, E.W., *Networks: Resource allocation.* Journal of Industrial Engineering, 1974. **6**(4): p. 22-32.
62. Herroelen, W.S., *Resource-constrained project scheduling -The state of the art* Operations Research Quarterly, 1972: p. 261-275.
63. Lakkhanawat, H. and M. Bagajewicz., *Financial Risk Management with Product Pricing in the Planning of Refinery Operations.* Ind. and Eng. Chem. Res. , 2008. **47**(17): p. 6622-6639
64. Woodward, D.G., *Life cycle costing theory, information acquisition and application.* International Journal of Project Management, 1997. **15**(6): p. 335-344.
65. Easterby-Smith, M., R. Thorpe, and P. Jackson, *Management Research*. 3rd Ed. ed. 2010, London: SAGE.
66. Bryman, A., *Social Research Methods*. 3rd edition ed. 2008, Oxford University Press.
67. Saunders, M., P. Lewis, and A. Thornhill, *Research methods for business students*. 4th ed ed. 2007, London: Prentice Hall.
68. Mangan, J., C. Lalwani, and B. Gardner, *Combining quantitative and qualitative methodologies in logistics research.* International Journal of Physical Distribution & Logistics Management, 2004. **34**(14).
69. Amaratunga, D., et al., *Quantitative and qualitative research in the built environment: application of "mixed" research approach".* Work Study, 2002. **51**(1): p. 17-31.
70. Saunders, M., P. Lewis, and A. Thornhill, *Research Methods for Business Students*. 2009: Pearson Education.
71. Saunders, M., P. Lewis, and A. Thornhill, *Research methods for business students*, 2009, Prentice Hall.: London.
72. Bhaskar, R.A., *Reclaiming Reality: A Critical Introduction to Contemporary Philosophy*. 1989, London: Verso.
73. Tashakkori, A. and C. Teddlie, *Mixed Methodology: Combining Qualitative and Quantitative Approaches*. 1998: SAGE publications.
74. K., N., Denzin, and Y.S. Lincoln, *The SAGE Handbook of Qualitative Research*. 2005: SAGE.
75. Thomas, D., *Diversity as Strategies.* Harvard Business Review, 2004. **September 2004**.
76. Hussey, J. and R. Hussey, *Business Research : A Practical Guide for Undergraduate and Postgraduate Students*. 1997: Basingstoke : Macmillan.
77. Ghauri, P. and K. Gronhaug, *Research Methods in Business Studies : A Practical Guide*. 2001: Harlow : Financial Times Prentice Hall.
78. Yin, R.K., *Case Study Research : Design and Methods*. 2003, Sage, London.

79. Myers, M.D., *Qualitative Research in Information Systems.* MIS Quarterly, 1997. **21**: p. 241-242.
80. Burney, P.D.S.M.A., *INDUCTIVE & DEDUCTIVE RESEARCH APPROACH.* Teaching material for MACE research postgraduate, 2008.
81. Hakim, C., *Research Design: Succesful Designs for Social Economics Research.* 2000.
82. Zikmund, W.G., *Business research methods*. 2003: Thomson/South-Western - Business & Economics
83. Balnaves, M. and P. Caputi, *Introduction to Quantitative Research Methods: An investigative approach.* 2001, London: Sage.
84. Malhotra, N.K. and M. Peterson, *Basic Marketing Research: A Decision-making Approach*. 2009: Pearson Education.
85. Balnaves, M. and P. Caputi, *Introduction to Quantitative Research Methods: An Investigative Approach*. 2001: SAGE.
86. Cornuejols, G. and R. TÄutÄuncÄu, *Optimization methods in Finance.* Carnegie Mellon University, Pittsburgh, 2005.
87. Biegler, L.T., *Nonlinear Programming: Concepts, Algorithms and Applications*. 2000: Chemical Engineering Department Carnegie Mellon University Pittsburgh, PA.
88. Mantell, J.B. and L.S. Lasdon, *A GRG ALGORITHM FOR ECONOMETRIC CONTROL PROBLEMS.* Annals of Economic and Social Measurement, 1977. **6**(5).
89. Michalewicz, Z. and M. Schoenauery, *Evolutionary Algorithms for Constrained Parameter Optimization Problems.* 2000.
90. PTT, *Annual report 2011*, 2011, PTT Public Company Limited.
91. PTT, *Annual Report 2012*, 2012, PTT Public Company Limited.
92. Herroelen, W. and B. De Reyck, *Phase transitions in project scheduling.* Journal of Operational Research Society, 1999. **50**: p. 148-156.
93. Burke, R., *Project Management planning&control techniques*. Third edition ed. 2006: John Wiley&sons LTD.
94. Camichael, D.G., *Project planning, and control*. 2006, Great Britain: Taylor&Francis Group
95. Lurin, P.A., *Business planning for managers* 2012: Investaura.
96. Blanchard, B.S. and W.J. Fabrycky, *Systems Engineering and Analysis*. 1990: Prentice-Hall.
97. Thesen, A., *Heuristic Scheduling of activities under resource and precedence restrictions.* Manage Sci, 1976. **23**(4).
98. Kolisch, R. and S. Hartmann, *Chapter 7*, in *Heuristic Algorithms for solving the resource-constrained project scheduling problem:classification and computational analysis*. 1998.
99. Davis, E.M., *Resource Allocation in Project Network Models-A Survey.* Industrial Engineering, 1966.
100. Mason, A.T. and C.L. Moodie, *A Branch and Bound Algorithm for Minimizing Cost in Project Scheduling.* Management Science, 1971. **18**(4): p. 158-173.

.

101. Bennington, G.E. and C.F. Mcginnis, *A Schedule Modifying Algorithm for Project Planning with Resource Constraints.* NSCO-IE Technical Report, 2000. **73**.
102. Wiest, T.D., *A Heuristic Model for Scheduling Large Projects with Limited Resources,.* Manage Sci, 1967. **13**(6): p. 359-377.
103. Blazewicz, J., J.K. Lenstra, and A.H.G. Rinnooy Kan, *Scheduling projects to resource constraints: classification and complexity.* Discrete Applied Mathematics, 1983. **5**: p. 11–24.
104. Kelley, J., *The critical path method: Resources planning and scheduling*. Industrial scheduling, ed. J.M.a.G. Thompson. 1963, New Jersey: Eds. Prentice Hall.
105. Kolisch, R. and R. Padman, *An integrated survey of deterministic project scheduling.* Omega The International Journal of Management Science, 2001. **29**: p. 249–272.
106. Brucker, P., et al., *Resource-constrained project scheduling: notation, classification, models, and methods.* . European Journal of Operational Research, 1999. **112**(1): p. 3–41.

107. Demeulemeester, E. and W. Herroelen, *New benchmark results for the resource constrained project scheduling problem.* Management Science, 1997. **43**(11): p. 1485-1492.
108. Mingozzi, A., et al., *An exact algorithm for project scheduling with resource-constraints based on a new mathematical formulation.* Manage Sci 1998. **44**(5): p. 714–29.
109. Sprecher, A., *Network decomposition techniques for resource-constrained project scheduling.* Journal of the Operational Research Society, 2002. **53**(4): p. 405-414.
110. Sprecher, A., R. Kolisch, and A. Drexl, *Semi active, active, and non delay schedules for the resource constrained project scheduling problem.* European Journal of Operational Research, 1995. **80**: p. 94-102.
111. Vanhoucke, M. *Optimizing regular scheduling objectives: Priority rule based scheduling*. 2012.
112. Demeulemeester, E., M. Vanhoucke, and W. Herroelen, *A random generator for activity-on-the-node networks.* Journal of Scheduling, 2003. **6**: p. 13-34.
113. Kolisch, R. and S. Hartmann, *Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis.* Publications of Lehrstuhl Fur Produktion Und Logistik, 1998.
114. Valls, V., F. Ballestin, and S. Quintanilla, *Justification and RCPSP: A technique that pays.* European Journal of Operational Research, 2005. **165**(2): p. 375-386.
115. Li, R.K.-Y. and J. Willis, *An iterative scheduling technique for resource-constrained project scheduling.* European Journal of Operational Research, 1992. **56**: p. 370-379.
116. Özdamar, L. and G. Ulusoy, *A note on an iterative forward/backward scheduling technique with reference to a procedure by Li and Willis.* . European Journal of Operational Research 1996. **89**: p. 400–7.
117. Özdamar, L. and G.A. Ulusoy, *local constraint based analysis approach to project scheduling under general resource constraints.* European Journal of Operational Research 1994. **79**: p. 287–98.
118. Elmaghraby, S.E., *Project bidding under deterministic and probabilistic activity durations.* European Journal of Operational Research, 1990. **49**: p. 14–34.
119. Alvarez-Valdés, R. and J.M. Tamarit, *Heuristic algorithms for resource-constrained project scheduling: a review and an empirical analysis. In: Slowiński R, Węglarz J, editors.* Advances in project scheduling. , 1989: p. 113–34.
120. Cooper, D., *Heuristics for scheduling resource constrained projects: An experimental investigation.* . Management Science, 1976. **22**(11): p. 1186-1194.
121. Boctor, F.F., *Heuristics for scheduling projects with resource restrictions and several resource-duration modes.* . International Journal of Production Research 1993. **31**(11): p. 2547–58.
122. Boctor, F.F., *A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes.* European Journal of Operational Research, 1996. **90**: p. 349–61.
123. Baroum, S.M. and J.H. Patterson, *An exact solution procedure for maximizing the net present value of cash flows in a network, in: Węglarz J (Ed.), Project scheduling-recent models, algorithms and applications.* Boston: Kluwer Academic Publishers 1999: p. 107–34.
124. Allam, S.I.G., *Multi-project scheduling: a new categorization for heuristic scheduling rules in construction scheduling problems.* Construction Management and Economics, 1988. **6**(2): p. 93-115.
125. Hartmann, S. and R. Kolisch, *Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem.* European Journal of Operational Research, 2000. **127**: p. 394-407.
126. Icmeli, O. and S.S. Erenguc, *A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows.* . Management Science 1996. **42**(10): p. 1395–408.
127. Kolisch, R., *Effcient priority rules for the resource-constrained project scheduling problem.* Journal of Operations Management 1996. **14**(3): p. 179-192.

128. Kolisch, R. and A. Sprecher, *PSPLIB - A project scheduling library.* European Journal of Operational Research Quarterly, 1997. **96**: p. 205-216.
129. Davis, E.W. and G.E. Heidorn, *An algorithm for optimal project scheduling under multiple resource constraints.* Management Science 1971. **17**: p. 803-816.
130. Kolisch, R., *Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation.* European Journal of Operational Research, 1996. **90**: p. 320-333.
131. Kurtulus, I. and E.W. Davis, *Multi-project scheduling: categorization of heuristic rules performance.* Management Science, 1982. **28**(2): p. 161–72.
132. Icmeli, O. and S. Erenguc, *A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows.* Computers & Operations Research 1994. **21**(8): p. 841–53.
133. Leung, J.-T., *Handbook of scheduling*. 2004: CRC press/Chapman & Hall.
134. Liu, S.S. and C.J. Wang, *Optimization model for resource assignment problems of linear construction projects.* Automation in Construction, 2007. **16**(4): p. 460–473.
135. Kirkpatrick, S., C.D. Gelatt jr., and M.P. Vecchi, *Optimization by simulated annealing.* Manage Sci, 1983. **220**: p. 671-680.
136. Martí, R., M. Laguna, and F. Glover, *Principles of scatter search.* European Journal of Operational Research, 2006. **169**(2): p. 359-372.
137. Glover, F., *Tabu search - Part I.* ORSA Journal on Computing, 1989. **1**: p. 190-206.
138. Glover, F., *Tabu search - Part II.* ORSA Journal on Computing, 1989. **2**: p. 4-32.
139. Mausser, H.E. and S.R. Lawrence, *Exploiting block structure to improve resource-constrained project schedules.* In: Glover, F., Osman, I. and Kelley, J. (Eds.), Metaheuristics 1995: State of the art, 1997.
140. Mastor, A., *An experimental and comparative evaluation of production line balancing techniques.* Management Science, 1970. **16**: p. 728-746.
141. Holland, H., *Adaptation in natural and articial systems.* . 1975, Ann Arbor: University ofMichigan Press.
142. Palmer, C. and A. Kershenbaum. *Representing trees in genetic algorithms*. in *Proceedings of the first IEEE International Conference on Evolutionary Computation*. 1994. New York.
143. Valls, V., F. Ballestín, and S. Quintanilla, *A hybrid genetic algorithm for the Resourceconstrained project scheduling problem with the peak crossover operator.* Eighth International Workshop on Project Management and Scheduling, 2002: p. 368-371.
144. Palpant, M., C. Artigues, and P. Michelon, *LSSPER: Solving the resource-constrained project scheduling problem with large neighbourhood search.* Annals of Operations Research, 2004. **131**: p. 237-257.
145. Pollack-Johnson, B., *Hybrid structures and improving forecasting and scheduling in project management.* . Journal of Operations Management 1995. **12**: p. 101–17.
146. Patterson, J.H., *Project scheduling: the effects of problem structure on heuristic scheduling.* Naval Research Logistics, 1976. **23**: p. 95-123.
147. Patterson, J.H., et al., *Computational experience with a backtracking algorithm for solving a general class of precedence and resource- constrained scheduling problems.* . European Journal of Operational Research, 1990. **49**: p. 68–79.
148. Patterson, J.H. and G.W. Roth, *Scheduling a project under multiple resource constraints: A zero–one programming approach.* AIIE Transactions 1979. **8**: p. 449-455.
149. Kolisch, R. and S. Hartmann, *Experimental investigation of heuristics for resource-constrained project scheduling: An update.* European Journal of Operational Research, 2006. **174**: p. 23–37.
150. Bell, C.E. and J. Han, *A new heuristic solution method in resource-constrained project scheduling.* . Naval Research Logistics Quarterly, 1991. **38**: p. 315–31.
151. Oguz, O. and H. Bala, *A comparative study of computational procedures for the resource constrained project scheduling problem.* . European Journal of Operational Research 1994. **72**: p. 406-416.

152. Pritsker, A., L. Watters, and P. Wolfe, *Multiproject scheduling with limited resources: A zero-one programming approach.* Management Science, 1969. **16**: p. 93-107.
153. Patterson, J.H. and W.D. Huber, *A horizon-varying, zero-one approach to project scheduling.* Manage Sci 1974. **20**(6): p. 990–8.
154. Stinson, J.P., E.W. Davis, and B.M. Khumawala, *Multiple resource-constrained scheduling using branch-and-bound.* AIIE Trans, 1978. **10**(3): p. 252–9.
155. Zhu, D. and R. Padman, *A cooperative multi-agent approach to constrained project scheduling.*, in *advances in Metaheuristics, optimization, and stochastic modeling technologies*, I.i.c.s.a.o. research, Editor. 1997, MA: Kluwer Academic Norwell. p. 367–81.
156. Tantisuvanichkul, V. and M. Kidd. *Project scheduling a review through literature*. in *RICS Construction and Property Conference (COBRA)*. 2011. University of Salford, Manchester, UK.
157. Tantisuvanichkul, V. and M. Kidd, *Max NPV a review through literature*, in *2nd International Conference on Construction and Project Management (ICCPM)*2011: Singapore.
158. Russell, A.H., *Cash flows in networks.* Management Science 1970. **16**(5): p. 357–73.
159. Grinold, R.C., *The payment scheduling problem.* Naval Research Logistics Quarterly, 1972. **19**(1): p. 123–36.
160. Elmaghraby, S.E. and W.S. Herroelen, *The scheduling of activities to maximize the net present value of projects.* European Journal of Operational Research 1990. **49**: p. 35-49.
161. Demeulemeester E, Herroelen W, and V.D. P., *An optimal recursive search procedure for the deterministic unconstrained max-npv project scheduling problem.*, 1996, Technical Report, Department of Applied Economics, Katholieke Universiteit Leuven.
162. Doersch, R.H. and J.H. Patterson, *Scheduling a project to maximize its present value: a zero–one programming approach.* Management Sciences 1977. **23**(8): p. 882–9.
163. Smith-Daniels, D.E. and V.L. Smith-Daniels, *Maximizing the net present value of a project subject to materials and capital constraints.* . Journal of Operations Management 1987. **7**: p. 33-45.
164. Tavares, L.V., *Multicriteria scheduling of a railway renewal program.* European Journal of Operational Research 1986. **25**: p. 395–405.
165. Baroum, S.M. and J.H. Patterson, *The development of cash flow weight procedures for maximizing the net present value of a project.* Journal of Operations Management 1996. **14**(3): p. 209–27.
166. Talbot, B. and J.H. Patterson, *An efficient integer programming algorithm with network cuts for solving resource- constrained scheduling problems.* Manage Sci, 1978. **24**: p. 1163-1174.
167. Yang, K.K., F.B. Talbot, and J.H. Patterson, *Scheduling a project to maximize its net present value: an integer programming approach.* European Journal of Operational Research 1992. **64**: p. 188–98.
168. Kolisch, R. and A. Drexl, *Adaptive search for solving hard project scheduling problems.* Naval Research Logistics Quarterly, 1996. **43**: p. 23–40.
169. Russell, R.A., *A comparison of heuristics for scheduling projects with cash flows and resource restrictions.* . Management Science 1986. **32**(10): p. 1291–300.
170. Padman, R., D.E. Smith-Daniels, and V.L. Smith-Daniels, *Heuristic scheduling of resource-constrained projects with cash flows.* Naval Research Logistics Quarterly, 1997. **44**: p. 364–81.
171. Padman, R. and D.E. Smith-Daniels, *Early-tardy cost trade-offs in resource constrained projects with cash flows: an optimization-guided heuristic approach.* . European Journal of Operational Research 1993. **64**: p. 295–311.
172. Zhu, D. and R. Padman, *A multiheuristic combination model for scheduling resource-constrained projects with cash flows.* , 1997, The Heinz School, Carnegie Mellon University: Pittsburgh.
173. Zhu, D. and R. Padman, *A metaheuristic scheduling procedure for resource-constrained projects with cash flows.* . Naval Research Logistics Quarterly, 1999. **46**: p. 1-18.

174. Smith-Daniels, D.E., R. Padman, and V.L. Smith-Daniels, *Heuristic scheduling of capital constrained projects.* Journal of Operations Management 1996. **14**(3): p. 241-54.
175. Erenguc, S.S., S. Tufekci, and C.J. Zappe, *The solution of the time/cost trade-off problem with discounted cash flows using generalized benders decomposition.*, in *Technical Report*, 1991, College of Business Administration, University of Florida,.
176. Ulusoy, G. and Ö. L., *A constraint-based perspective in resource constrained project scheduling.* International Journal of Production Research 1994. **32**(3): p. 693–705.
177. Kolisch, R. and A. Drexl, *Local search for non preemptive multi-mode resource-constrained project scheduling.* . IIE Transactions 1997. **29**(11): p. 987–99.
178. Dayanand, N. and R. Padman, *A simulated annealing approach for scheduling payments in projects.*, 1994, The Heinz School, Carnegie Mellon University, Pittsburgh.
179. Tormos, P. and A. Lova, *A Competitive Heuristic Solution Technique for Resource-Constrained Project Scheduling.* Annals of Operations Research, 2001. **102**: p. 65–81.
180. Aquilano, N.J. and D.E. Smith, *A Formal Set of Algorithms for Project Scheduling with Critical Path Method Material Requirements Planning.* Journal of Operations Management, 1980. **1**(2).
181. Smith-Daniels, D.E. and N.J. Aquilano, *Constrained resource project scheduling subject to material constraints.* Journal of Operations Management, 1984. **4**(4).
182. Birbil, S.I. and S.C. Fang, *An electromagnetism-like mechanism for global optimization.* Journal of Global Optimization, 2003. **25**: p. 263-282.
183. Balas, E., *Project scheduling with resource constraints*, in *Applications of Mathematical Programming techniques*, i.E.M.L.B. (Ed.), Editor. 1971, English University Press: London. p. 187- 200.
184. F B Talbot and J.H. Patterson, *An efficient integer programming algorithm with network cuts for solving resource- constrained scheduling problems.* Management Science 1978, 1978. **24**(11): p. 1163–74.
185. Demeulemeester, E. and W.S. Herroelen, *A branch- and-bound procedure for the multiple resource-constrained project scheduling problem with generalize precedence constrained.* European Journal of Operational Research, 1992. **111**: p. 152-174.
186. Herroelen, W., De Reyck, B. and Demeulemeester, E. , *Resource-constrained project scheduling: a survey of recent developments.* Computers and Operations Research, 1998. **25**(4): p. 279–302.
187. Debels, D., et al., *A hybrid scatter-search/electromagnetism meta-heuristic for the resource-constrained project scheduling problem.* European Journal of Operational Research, 2006. **169**(3): p. 638-653.
188. De Reyck, B.a.H., W. , *An optimal procedure for the resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations.* Computers and Operations Research, 1998. **25**(1): p. 1–17.
189. Klein, R., *Bidirectional planning — improving priority rule based heuristics for scheduling resource-constrained projects.* . **Technical Report**, 1998.
190. Pinder, J.P. and A.S. Marucheck, *Using discounted cash flow heuristics to improve project net present value.* Journal of Operations Management, 1996. **14**(3): p. 229–40.
191. Tantisuvanichkul, V. and M. Kidd. *Improve Net Present Value using cash flow weight*. in *2nd International Conference on Construction and Project Management (ICCPM)*. 2011. Singapore.
192. Tsai, D.M. and H.N. Chiu, *Two heuristics for scheduling multiple projects with resource constraints.* . Construction Management and Economics, 1996. **14**(6): p. 325–40.
193. Valls, V., S. Quintanilla, and F. Ballestín, *Resource-constrained project scheduling: a critical activity reordering heuristic.* European Journal of Operational Research, 2003. **149**: p. 282-301.

Appendices

Appendix A: Numerical Illustration MINSLK/m-CCF

Full numerical illustration on comparing MINSLK and m-CCF rule from Chapter 6.2.1

Table 1 Activity Data with slack and m-CCF

Activity	Cash flow	Predecessors	Duration	Total Slack
1	-20	-	2	0
2	60	-	4	5
3	55	-	2	6
4	-10	1	4	0
5	50	4	2	0
6	-40	2, 4	1	3
7	35	3, 5	2	0

Network diagram adapted from Russell's [169]. The activities in red denote the activities in critical path. Capital Constraint = -5.

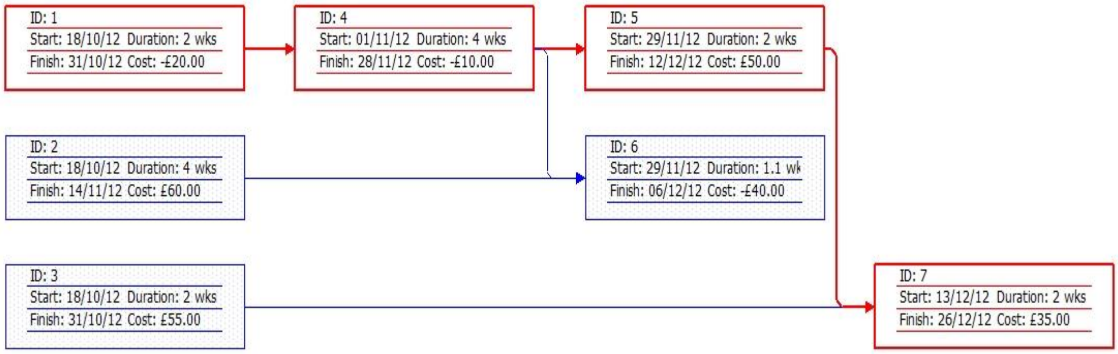


Figure 1 Network Diagrams for MINSLK and m-CCF comparison

According to [m-CCF]-SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown Table 2.

Table 2 m-CCF values for each activity

Task						mCCF
1	-20	-9.607894392	44.34602184	-34.08575156	29.82503261	10.4774085
2	60	-34.08575156				25.91424844
3	55	32.30907212				87.30907212
4	-10	46.15581732	-35.47681747	31.04221529		31.72121514
5	50	33.62763037				83.62763037
6	-40					-40
7	35					35

From

$$m - CCF_j = \max[CF_j \exp(-\propto t_j) + \sum_{k \in S_j} CF_k \exp(-\propto (t_k - t_j))] \qquad (5.6)$$

Demonstration

[1] = -20 + (-10)exp(-2α) + 50 exp(-6α) + (-40) exp(-8α) + 35exp(-8α)

[2] = 60 + (-40)exp(-8α)

[3] = 55 + 35exp(-4α)

[4] = -10 + 50exp(-4α) + (-40) exp(-6α) + 35exp(-6α)

[5] = 50 + 35exp(-2α)

[6] = -40

[7] = 35

The schedule can be done by applying both rules in serial generation scheme coded as results shown in Table 3 and 4.

Table 3 The output from applying MINSLK rule

g	D_g	j
1	[1,2,3]	1,2
2	[3,4]	4
3	[3]	3
4	[5,6]	5
5	[6,7]	6,7

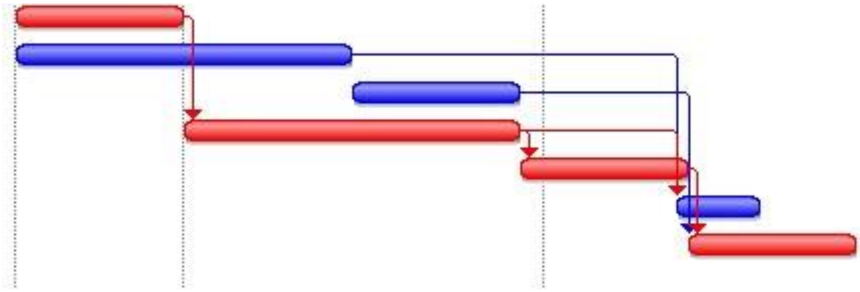
Table 4 The output from applying m-CCF rule

g	D_g	j
1	[1,2,3]	1,3
2	[2,4]	2,4
3	[5]	5
4	[6,7]	6,7

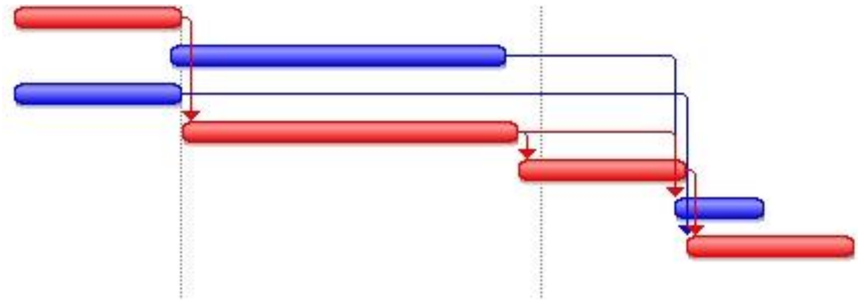
MINSLK put the activities in an increasing order of their slack value in the list that results in the schedule as shown in Figure 2(a).

m-CCF put the activities in an increasing order of their m-CCF value (display in Table 2) and the schedule is shown in Figure 2(b).

Among the three non-critical activities, in this case 2, 3, and 6, MINSLK give priority to activity 6, 2 and 3. m-CCF gives priority to 3, 2 and 6 respectively.



(a) MINSLK's rule



(b) m-CCF rule

Figure 2 The schedule from (a) MINSLK rule (b) m-CCF rule

NPV are then calculated according to each rule as shown in Table 5 and 6.

Table 5 NPV obtained from MINSLK rule

Activity	NewTs	tj	PV
1	1	0	-20
2	1	0	60
3	5	4	50.77139905
4	3	2	-9.607894392
5	7	6	44.34602184
6	9	8	-34.08575156
7	9	8	29.82503261
		NPV	121.2488076

NPV = -20+60+55exp(-4α)+(-10)exp(-2α) + 50exp(-6α)+(-40) exp(-8α) + 35exp(-8α)

NPV obtained from MINSLK rule = 121.2488076.

Table 6 NPV obtained from m-CCF rule

Activity	New Ts	tj	PV
1	1	0	-20
2	3	2	57.64736635
3	1	0	55
4	3	2	-9.607894392
5	7	6	44.34602184
6	9	8	-34.08575156
7	9	8	29.82503261
		NPV	123.1247748

NPV = -20+60exp(-2α)+55+(-10)exp(-2α)+50exp(-6α) +(-40) exp(-8α) + 35exp(-8α)

NPV obtained from m-CFF rule = 123.1247748.

Appendix B: Numerical Illustration GNS/m-CCF

Full numerical illustration on comparing GNS and m-CCF rule from Chapter 6.2.2

Table 1 Activity Data with successor and m-CCF

Activity	Cash flow	Predecessors	Duration	Successors
1	-200	-	7	2, 5, 8
2	-200	1	5	5, 8
3	350	-	1	4, 5, 8
4	100	3	4	5, 8
5	250	2, 4	2	8
6	300	-	6	7, 8
7	150	6	1	8
8	200	5, 7	1	-

Network diagram adapted from Padman's [170]. The activities in red denote the activities in critical path. Constraint = -100

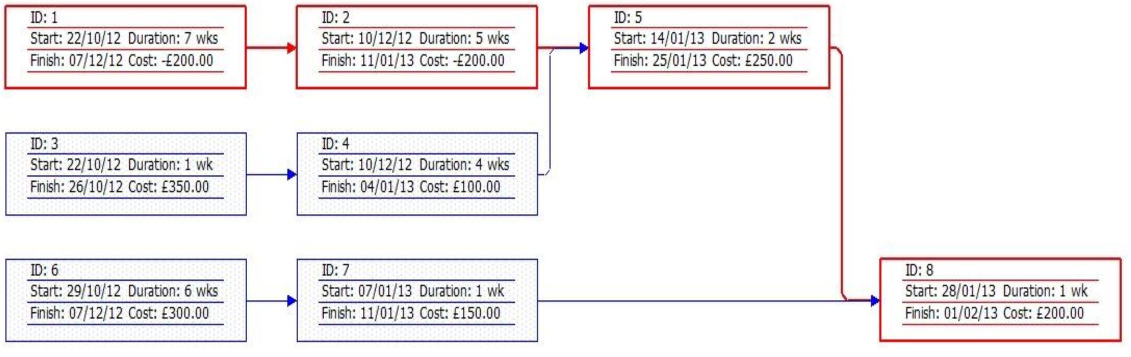


Figure 1 Network diagram for GNS and m-CCF comparison.

According to [m-CCF]-SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown in Table 2.

Table 2 m-CCF values for each activity

Task					mCCFW
1	-200	-173.8716471	196.6569653	151.1567483	-26.05793352
2	-200	226.2093545	173.8716471		200.0810016
3	350	98.01986733	196.6569653	151.1567483	795.8335809
4	100	200.6296995	154.2103172		454.8400167
5	250	192.1578878			442.1578878
6	300	133.0380655	151.1567483		584.1948138
7	150	170.4287578			320.4287578
8	200	0			200

From

$$m - CCF_j = \max[CF_j \exp(-\propto t_j) + \sum_{k \in S_j} CF_k \exp(-\propto (t_k - t_j))] \qquad (5.6)$$

Demonstration

[1] = -200 + (-200)exp(-7α) +250exp(-12α) + 200exp(-14α)

[2] = -200 + 250exp(-5α) +200exp(-7α)

[3] = 350 + 100exp(-α) +250exp(-12α) + 200exp(-14α)

[4] = 100 +250exp(-11α) + 200exp(-13α)

[5] = 250 + 200exp(-2α)

[6] = 300+ 150exp(-6α) +200exp(-14α)

[7] = 150 + 200exp(-8α)

[8] = 200

The schedule can be done by applying both rules in serial generation scheme as results shown in Table 3 and 4.

Table 3 The output from applying GNS rule

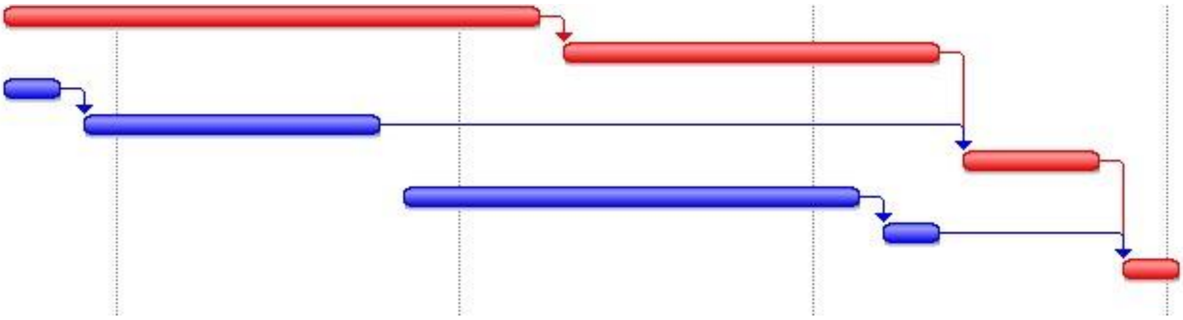
g	D_g	j
1	[1,3,6]	1,3
2	[4,6]	4
3	[6]	6
4	[2,7]	2
5	[7]	7
6	[5]	5
7	[8]	8

Table 4 The output from applying m-CCF rule

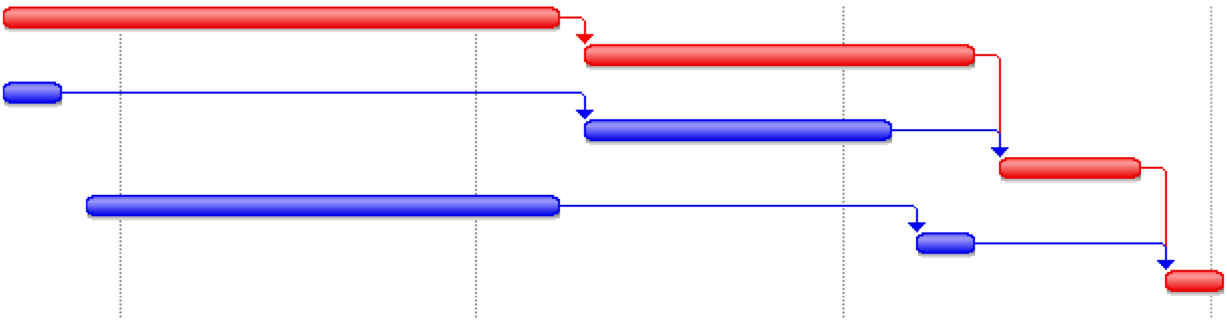
g	D_g	j
1	[1,3,6]	1,3
2	[4,6]	6
3	[2,4,7]	2,4
4	[7]	7
5	[5]	5
6	[8]	8

GNS put the activities in an increasing order of their number of successors in the schedule as shown in Figure 2(a).

m-CCF put the activities in an increasing order of their m-CCF value (display in Table 2) that results is shown in Figure 2(b).



(a) GNS rule



(b) m-CCF results

Figure 2 The schedule from (a) GNS rule (b) m-CCF rule

Among the four non-critical activities, in this case 3, 4, 6 and 7, GNS gives priority to activity 3, 4, 6 and 7 respectively. m-CCF gives priority to 3, 6, 4 and 7 respectively.

NPV are then calculated according to each rule as shown in Table 5 and 6.

Table 5 NPV obtained from GNS rule

Activity	NewTs	tj	PV
1	1	0	-200
2	8	7	-173.8716471
3	1	0	350
4	2	1	98.01986733
5	13	12	196.6569653
6	6	5	271.4512254
7	12	11	120.3778197
8	15	14	151.1567483
		NPV	662.6342306

NPV $= -200+(-200)\exp(-7\alpha)+350+100\exp(-\alpha)+250\exp(-12\alpha)+300\exp(-5\alpha) + 150\exp(-11\alpha)+ 200\exp(-14\alpha)$

NPV obtained from GNS rule = 662.6342306.

Table 6 NPV obtained from m-CCF rule

Activity	New Ts	tj	PV
1	1	0	-200
2	8	7	-173.8716471
3	1	0	350
4	8	7	86.93582354
5	13	12	196.6569653
6	2	1	294.059602
7	9	8	127.8215683
8	15	14	151.1567483
		NPV	832.7590604

NPV $= -200+(-200)\exp(-7\alpha)+350+100\exp(-7\alpha)+250\exp(-12\alpha)+300\exp(-\alpha) + 150\exp(-8\alpha)+ 200\exp(-14\alpha)$

NPV obtained from [m-CFF] rule = 832.7590604.

Appendix C: Numerical Illustration CCF/m-CCF

Full numerical illustration on comparing CCF and m-CCF rule from Chapter 6.2.3

Table 1 Activity Data with CCF and m-CCF adapted from Baroum's Example [165]

Activity	Cash flow	Predecessors	Duration
1	300	-	1
2	521	1	4
3	15	2,8	2
4	310	-	6
5	535	4	1
6	-10	3,5	1
7	-300	-	6
8	-20	7	6

Network Diagram adapted from Baroum's Example [165]. The activities in red denote the activities in critical path. Capital Constraint= -10

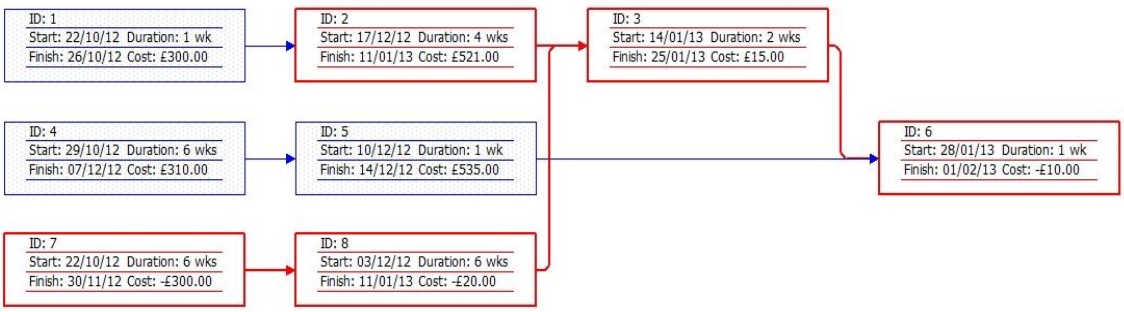


Figure 1 Network diagram for CCF and m-CCF comparison

From equation (5.5), the CCF values are determined as shown in Table 2.

Table 2 CCF value obtain for each activity

Task	successors	Cash Flow				CCF
1	2,3,6	300	521	15	-10	826
2	3,6	521	15	-10		526
3	6	15	-10			5
4	5,6	310	535	-10		835
5	6	535	-10			525
6		-10				-10
7	8,3,6	-300	15	-10	-20	-315
8	3,6	-20	15	-10		-15

From

$$CCF_j = \max(CF_j + \sum_{k \in S_j} CF_k) \qquad (5.16)$$

Demonstration

[1] = 300+521+15+(-10) = 826

[2] = 521+15+(-10) = 526

[3] = 15+(-10) = 5

[4] = 310+535+(-10) = 835

[5] = 535+(-10) = 525

[6] = -10

[7] = -300+15+(-10)+(-20) = -315

[8] = -20+15+(-10) = -15

According to [m-CCF]-SSGS procedure and flow chart (Figure 5.1), the m-CCF values are determined as shown in Table 3.

Table 3 m-CCF values obtained for each activity

Task					mCCF
1	300	500.5712978	11.79941792	-7.557837415	804.8128783
2	521	12.2809613	-7.866278611		525.4146827
3	15	-9.607894392			5.392105608
4	310	474.5024336	-7.557837415		776.9445962
5	535	-8.52143789			526.4785621
6	-10				-10
7	-300	11.79941792	-7.557837415	-17.73840873	-313.4968282
8	-20	13.30380655	-8.52143789		-15.21763134

From

$$m - CCF_j = \max[CF_j \exp(-\propto t_j) + \sum_{k \in S_j} CF_k \exp(-\propto (t_k - t_j))] \qquad (5.6)$$

Demonstration

[1] = 300 + 521exp(-2α) +15exp(-12α) + (-10)exp(-14α)

[2] = 521 + 15exp(-10α) +(-10)exp(-12α)

[3] = 15 + (-10)exp(-2α)

[4] = 310 + 535exp(-6α) + (-10)exp(-14α)

[5] = 535 + (-10)exp(-8α)

[6] = -10

[7] = -300 + 15exp(-12α) +(-10)exp(-14α) + (-20)exp(-6α)

[8] = -20 + 15exp(-6α) +(-10)exp(-8α)

The schedule can be done by applying both rules in serial generation scheme as results shown in Table 4 and 5.

Table 4 The output from applying CCF rule

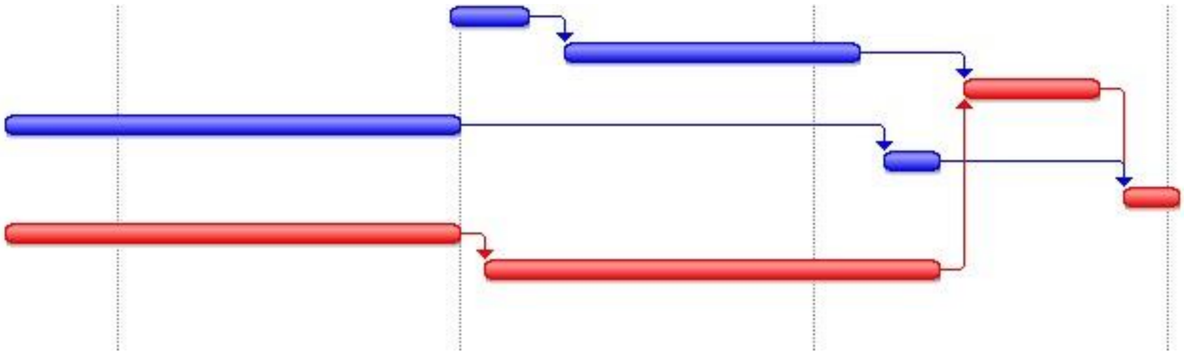
g	D_g	j
1	[1,4,7]	4,7
2	[1,5,8]	1,8
3	[2,5]	2
4	[5]	5
5	[3]	3
6	[6]	6

Table 5 The output from applying m-CCF rule

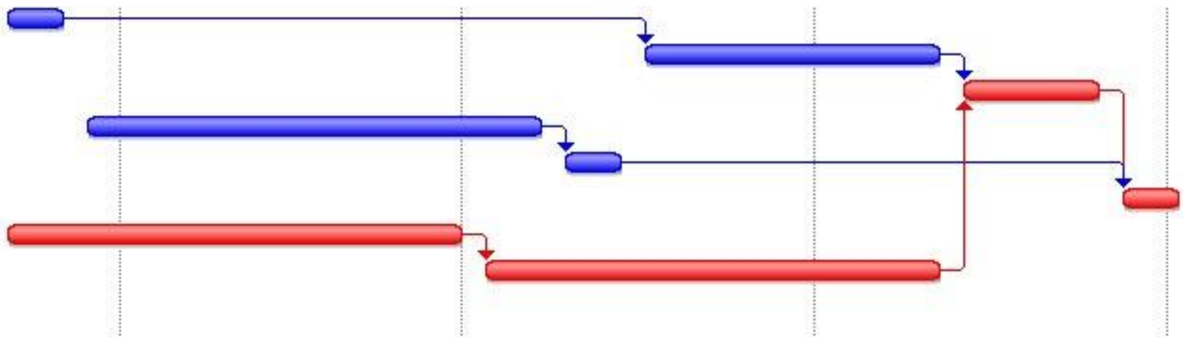
g	D_g	j
1	[1,4,7]	1,7
2	[2,4]	4
3	[2,5,8]	8
4	[2,5]	5
5	[2]	2
6	[6]	6
7	[8]	8

CCF put the activities in an increasing order of their CCF value (display in Table 2) that results in the schedule as shown in Figure 2(a).

m-CCF put the activities in an increasing order of their m-CCF value (display in Table 3) that results in the schedule as shown in Figure 2(b).



(a) CCF rule



(b) m-CCF rule

Figure 2 The schedule from (a) CCF rule (b) m-CCF rule

In this case, the non critical activities are activity 1, 2, 4, and 5.

CCF Give priority to activity 4, 1, 2 and 5. so they schedule activity 4 first by shift to the right as much as possible followed by activity 1, 2 and 5.

m-CCF gives priority to 1, 4, 5 and 2 respectively. So they schedule activity 1 first by shift to the right as much as possible followed by activity 4 and 5.

NPV are then calculated according to each rule as shown in Table 6 and 7.

Table 6 NPV obtained from CCF rule

Activity	NewTs	tj	PV
1	7	6	266.076131
2	8	7	452.9356406
3	13	12	11.79941792
4	1	0	310
5	12	11	429.3475569
6	15	14	-7.557837415
7	1	0	-300
8	7	6	-17.73840873
		NPV	1162.600909

NPV = 300exp(-6α)+521exp(-7α) +15exp(-12α) + 310 + 535exp(-11α) +(-10)exp(-14α) + (-300)+(-20)exp(-6α)

NPV obtained from CCF = 1162.600909.

Table 7 NPV obtained from m-CCF rule

Activity	New Ts	tj	PV
1	1	0	300
2	9	8	443.9669141
3	13	12	11.79941792
4	2	1	303.8615887
5	8	7	465.1066559
6	15	14	-7.557837415
7	1	0	-300
8	7	6	-17.73840873
		NPV	1199.43833

NPV = 300 + 521exp(-8α) + 15exp(-12α) + 310 exp(-α) + 535exp(-7α) + (-10)exp(-14α) + (-300)+ (-20)exp(-6α)

NPV obtained from m-CCF rule = 1199.43833.

Appendix D: The Serial and Parallel results

The Full Results from 7.2.1

1 Activity Data generated

Task ID	Start	Finish	Cash Flow	Predecessors
1	24/10/2012	26/10/2012	90	
2	29/10/2012	30/10/2012	80	
3	31/10/2012	01/11/2012	-10	
4	02/11/2012	05/11/2012	86	3
5	29/10/2012	01/11/2012	40	4
6	02/11/2012	15/11/2012	56	5
7	29/10/2012	29/10/2012	70	3
8	16/11/2012	19/11/2012	75	7
9	12/11/2012	15/11/2012	88	3
10	08/11/2012	09/11/2012	102	
11	20/11/2012	23/11/2012	20	8
12	26/11/2012	27/11/2012	-30	
13	26/11/2012	26/11/2012	-60	
14	28/11/2012	29/11/2012	-20	11
15	30/11/2012	06/12/2012	-48	14
16	23/11/2012	29/11/2012	45	12,14
17	07/12/2012	12/12/2012	48	13,15
18	13/12/2012	21/12/2012	55	
19	24/12/2012	25/12/2012	61	17
20	24/12/2012	28/12/2012	71	
21	24/12/2012	28/12/2012	81	19,20
22	31/12/2012	03/01/2013	90	21
23	04/01/2013	14/01/2013	-20	22
24	15/01/2013	16/01/2013	-10	22
25	15/01/2013	21/01/2013	-15	22
26	15/01/2013	21/01/2013	-35	24,25
27	22/01/2013	25/01/2013	-35	26
28	28/01/2013	05/02/2013	-55	27
29	06/02/2013	07/02/2013	16	27
30	26/12/2012	31/12/2012	40	27
31	26/12/2012	01/01/2013	62	29,30
32	26/12/2012	01/01/2013	90	31
33	26/12/2012	31/12/2012	40	32
34	02/01/2013	07/01/2013	90	23
35	08/01/2013	16/01/2013	-50	23
36	17/01/2013	18/01/2013	-60	23
37	21/01/2013	29/01/2013	-40	23
38	30/01/2013	05/02/2013	-50	34,35,36,37
39	21/01/2013	21/01/2013	-38	38

40	22/01/2013	25/01/2013	56	39
41	06/02/2013	07/02/2013	64	
42	28/01/2013	04/02/2013	77	40
43	05/02/2013	05/02/2013	32	42
44	08/02/2013	08/02/2013	-41	
45	08/02/2013	08/02/2013	-58	44
46	11/02/2013	14/02/2013	20	43,45
47	27/02/2013	28/02/2013	29	
48	08/02/2013	15/02/2013	60	47
49	18/02/2013	18/02/2013	50	46,48
50	01/03/2013	01/03/2013	40	
51	11/02/2013	18/02/2013	60	33
52	19/02/2013	26/02/2013	50	51

2. NPV obtained from Schedule with m-CCF rule using optimal solution

Task ID	PV	GRG
1		90
2		72.87326463
3		-1.418457854
4		72.24761116
5		36.67976791
6		47.18950482
7		63.82489045
8		47.58296701
9		60.45756653
10		75.88941582
11		11.85666202
12		-15.34687958
13		-7.761807085
14		-9.785244008
15		-22.76626642
16		24.87541279
17		20.01176288
18		20.31374935
19		18.06080789
20		21.01310956
21		23.96541123
22		23.13859512
23		-4.009519642
24		-0.991266226
25		-2.412088727
26		-5.652603746

27		-4.916215751
28		-6.860283409
29		1.968209219
30		11.39394867
31		17.63433726
32		25.57665
33		11.39394867
34		22.22984386
35		-10.90718363
36		-10.94118996
37		-6.728635136
38		-7.031136765
39		-5.556380444
40		9.2729662
41		7.846117776
42		11.30150184
43		4.007036634
44		-4.81562695
45		-5.808006939
46		2.223355084
47		2.337173568
48		71.4150672
49		48.7149303
50		31.2768318
51		67.2110867
52		47.2101116
	NPV	726.1150759

3. NPV obtained from Schedule with m-CCF rule

Task ID	PV	SSGS	PSGS
1		90	90
2		67.74362459	72.87326463
3		-1.420386044	-8.24469747
4		67.20581907	72.24761116
5		36.67976791	36.67976791
6		47.18950482	47.18950482
7		59.89188259	63.82489045
8		47.58296701	47.58296701
9		56.92935102	60.45756653
10		70.32564524	75.88941582
11		11.85666202	11.85666202

12		-15.34687958	-15.3468795
13		-7.819477479	-30.8524196
14		-9.785244008	-9.78524400
15		-22.76626642	-22.7662664
16		24.28010332	24.87541279
17		20.01176288	20.01176288
18		20.31374935	20.31374935
19		18.06080789	18.06080789
20		20.58852044	21.01310956
21		23.41292372	23.96541123
22		22.62362594	23.13859512
23		-4.024919101	-4.70521488
24		-0.992207963	-1.87991744
25		-2.417663523	-2.83061234
26		-5.683201586	-6.63339194
27		-4.939363719	-5.76923262
28		-6.905342557	-8.05061715
29		1.964494668	1.968209219
30		11.26928092	11.39394867
31		17.63433726	17.63433726
32		25.57665	25.57665
33		11.26928092	11.39394867
34		22.22984386	22.22984386
35		-10.90718363	-10.9071836
36		-10.94118996	-10.9411899
37		-6.728635136	-6.72863513
38		-7.031136765	-7.03113676
39		-5.585945942	-6.39135884
40		9.190418567	9.2729662
41		7.846117776	7.846117776
42		11.17885063	11.30150184
43		3.99163541	4.007036634
44		-4.81562695	-4.81562695
45		-5.840309428	-6.81575928
46		2.218614851	2.223355084
47		2.337173568	2.337173568
48		0.713661735	0.714150672
49		48.6921804	0.487149303
50		31.2768318	31.2768318
51		67.2110867	67.2110867
52		47.2101116	54.968543
	NPV	700.1100003	689.406068

Appendix E: Forward-Backward Results

The Full Results from 7.2.2

1. NPV obtained m-CCF rule with Forward straetgy

Task ID	PV	SSGS	PSGS
1		90	90
2		67.74362459	72.87326463
3		-1.420386044	-8.24469747
4		67.20581907	72.24761116
5		36.67976791	36.67976791
6		47.18950482	47.18950482
7		59.89188259	63.82489045
8		47.58296701	47.58296701
9		56.92935102	60.45756653
10		70.32564524	75.88941582
11		11.85666202	11.85666202
12		-15.34687958	-15.3468795
13		-7.819477479	-30.8524196
14		-9.785244008	-9.78524400
15		-22.76626642	-22.7662664
16		24.28010332	24.87541279
17		20.01176288	20.01176288
18		20.31374935	20.31374935
19		18.06080789	18.06080789
20		20.58852044	21.01310956
21		23.41292372	23.96541123
22		22.62362594	23.13859512
23		-4.024919101	-4.70521488
24		-0.992207963	-1.87991744
25		-2.417663523	-2.83061234
26		-5.683201586	-6.63339194
27		-4.939363719	-5.76923262
28		-6.905342557	-8.05061715
29		1.964494668	1.968209219
30		11.26928092	11.39394867
31		17.63433726	17.63433726
32		25.57665	25.57665
33		11.26928092	11.39394867
34		22.22984386	22.22984386
35		-10.90718363	-10.9071836
36		-10.94118996	-10.9411899
37		-6.728635136	-6.72863513
38		-7.031136765	-7.03113676

39		-5.585945942	-6.39135884
40		9.190418567	9.2729662
41		7.846117776	7.846117776
42		11.17885063	11.30150184
43		3.99163541	4.007036634
44		-4.81562695	-4.81562695
45		-5.840309428	-6.81575928
46		2.218614851	2.223355084
47		2.337173568	2.337173568
48		71.3661735	71.4150672
49		48.6921804	48.7149303
50		31.2768318	31.2768318
51		67.2110867	67.2110867
52		47.2101116	54.968512
	NPV	700.1100003	689.406068

2. NPV obtained m-CCF rule Backward straetegy

Task ID	PV	SSGS	PSGS
1		90	90
2		64.49592446	57.74258585
3		-1.421849904	-1.424972646
4		64.00886193	57.35884283
5		36.67976791	36.67976791
6		47.18950482	47.18950482
7		57.34621374	52.0280429
8		47.58296701	47.58296701
9		54.62684036	49.80870209
10		66.82890592	59.69672885
11		11.85666202	11.85666202
12		-15.34687958	-15.34687958
13		-7.863945776	-7.959335898
14		-9.785244008	-9.785244008
15		-22.76626642	-22.76626642
16		23.85631745	22.95689546
17		20.01176288	20.01176288
18		20.31374935	20.31374935
19		18.06080789	18.06080789
20		20.28339718	19.63511502
21		23.01874539	22.18193908
22		22.2554717	21.47372777
23		-4.03668463	-4.061787677
24		-0.992922171	-0.994445681

25		-2.421906158	-2.430957342
26		-5.706673285	-5.652603746
27		-4.957088593	-4.916215751
28		-6.940009967	-6.861966823
29		1.961697907	1.95573377
30		11.17755709	11.16833337
31		17.63433726	17.63433726
32		25.57665	25.57665
33		11.17755709	11.16833337
34		22.22984386	22.22984386
35		-10.90718363	-10.90718363
36		-10.94118996	-10.94118996
37		-6.728635136	-6.728635136
38		-7.031136765	-7.031136765
39		-5.608620379	-5.556380444
40		9.129368347	8.999316586
41		7.846117776	7.846117776
42		11.08859004	10.89638549
43		3.980097256	3.955498134
44		-4.81562695	-4.81562695
45		-5.865098186	-5.808006939
46		2.215048061	2.207442051
47		2.337173568	2.337173568
48		71.3292473	71.2504908
49		48.6749892	48.638323
50		31.2768318	31.2768318
51		67.2110867	67.2110867
52		47.2101116	47.2101116
	NPV	683.2899994	649.225856

3. NPV obtained optimal model

Task ID	PV	GRG
1		90
2		72.87326463
3		-1.418457854
4		72.24761116
5		36.67976791
6		47.18950482
7		63.82489045
8		47.58296701
9		60.45756653
10		75.88941582
11		11.85666202

12		-15.34687958
13		-7.761807085
14		-9.785244008
15		-22.76626642
16		24.87541279
17		20.01176288
18		20.31374935
19		18.06080789
20		21.01310956
21		23.96541123
22		23.13859512
23		-4.009519642
24		-0.991266226
25		-2.412088727
26		-5.652603746
27		-4.916215751
28		-6.860283409
29		1.968209219
30		11.39394867
31		17.63433726
32		25.57665
33		11.39394867
34		22.22984386
35		-10.90718363
36		-10.94118996
37		-6.728635136
38		-7.031136765
39		-5.556380444
40		9.2729662
41		7.846117776
42		11.30150184
43		4.007036634
44		-4.81562695
45		-5.808006939
46		2.223355084
47		2.337173568
48		71.4150672
49		48.7149303
50		31.2768318
51		67.2110867
52		47.2101116
	NPV	726.1150759

Appendix G: Project 1 Results

PTT Project		
Dates		
Investment date	3/1/11	
First Quarter End	3/31/11	
COD months after inv.	10	
COD Date	1/31/12	
First quarter after COD	3/31/12	
Project Life	10	years
Last quarter	3/31/22	
Financing		
Interest Rate	7.00%	pa.
Loan Term	10	year
WACC	9.16%	
Capex		
Land	15	MM Baht
EPC	600	MM Baht
Pre-investment	5	MM Baht
Contingency	3%	
	18	MM Baht
Total Costs	638	MM Baht
Life of Asset	10	year
Inverter Change	-	MM Baht
at (year)	-	
Life of Asset	-	year
Exchange Rate Assumption		
1 USD =	29.96	THB
1 Euro =	40.81	THB
1 USD =	0.73	Euro
Project Cost and Financing		
Investment Cost	638.00	MM Baht
IDC	21.65	
Total Cost	659.65	
Debt	462	MM Baht
% of Debt	70%	
Equity	198	MM Baht
% of Equity	30%	
Tenor	10	years after COD
Moratorium	0	years after COD
Moratorium start	3/31/12	
Moratorium End	3/31/12	
First repayment date	6/30/12	
Last repayment date	6/30/22	

Life Time (Year)		1	2	3	4	5	6	7	8	9	10
Total Revenue		167.07	183.10	183.99	184.97	186.03	187.02	188.01	189.09	190.28	191.59
CER Revenue		-	-	-	-	-	-	-	-	-	-
Cost of Good Sold		-	-	-	-	-	-	-	-	-	-
Selling & Admin. Cost		13.74	15.40	15.66	15.93	16.20	16.48	16.76	17.06	17.36	17.67
EBITDA		153.32	167.69	168.33	169.04	169.83	170.54	171.24	172.03	172.92	173.92
Depreciation		(59.05)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)
EBIT		94.277	103.279	103.914	104.624	105.415	106.126	106.828	107.617	108.509	109.509
Net income after TAX		94.28	103.28	103.91	104.62	105.42	106.13	106.83	92.87	93.14	93.51
Add back depreciation		59.05	64.41	64.41	64.41	64.41	64.41	64.41	64.41	64.41	64.41
Net Cash Flow	**-659.65**	**153.32**	**167.69**	**168.33**	**169.04**	**169.83**	**170.54**	**171.24**	**157.28**	**157.56**	**157.92**
Interest Repayment		(31.80)	(28.68)	(25.45)	(22.21)	(19.04)	(15.75)	(12.52)	(9.29)	(6.07)	(2.82)
Principal Repayment		(34.63)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)
Cash flow Available Before Dividends Payment		**86.89**	**92.84**	**96.71**	**100.65**	**104.62**	**108.61**	**112.55**	**101.82**	**105.31**	**108.92**
Discount Factor		1.0916	1.1916	1.3007	1.4199	1.5499	1.6919	1.8469	2.0161	2.2008	2.4023
PV of CF	**-659.65**	**140.46**	**140.73**	**129.41**	**119.05**	**109.57**	**100.80**	**92.72**	**78.01**	**71.59**	**65.74**
Cumulative CF		**(506.33)**	**(338.64)**	**(170.31)**	**(1.27)**	**168.56**	**339.10**	**510.34**	**667.63**	**825.18**	**983.10**

Investment Measures	
NPV	**388.42**
NPV on Equity	**441.18**
IRR	**21.29%**

1. Forward strategy, F-SSGS and F-PSGS

Task ID	SSGS	PSGS	Task ID	SSGS	PSGS
1	-5	-5	53	-13.26261742	-11.22221474
2	-9.320557864	-8.603591874	54	-6.12120804	-12.24241608
3	-1.953322389	-1.502555684	55	-15.3030201	-14.28281876
4	-9.283931304	-3.315689751	56	-10.2020134	-15.3030201
5	-8.143536762	-9.953211598	57	-7.14140938	-9.18181206
6	-11.69378296	-8.352702114	58	-6.12120804	-11.22221474
7	-3.687974995	-4.425569994	59	-11.22221474	-5.1010067
8	-9.469254683	-8.206687392	60	-11.22221474	-15.3030201
9	-6.760890554	-6.760890554	61	-14.28281876	-7.14140938
10	-2.913741262	-2.913741262	62	-7.14140938	-6.12120804
11	-3.496489514	-6.410230776	63	-8.16161072	-6.12120804
12	-3.617959341	-7.235918683	64	-12.24241608	-7.14140938
13	-2.000066817	-1.733391242	65	-7.14140938	-9.18181206
14	-5.959023645	-3.476097127	66	-13.26261742	-13.26261742
15	-2.862683493	-5.725366986	67	-5.1010067	-11.22221474
16	-3.038907084	-7.597267709	68	-14.28281876	-9.18181206
17	-5.806960764	-4.147829117	69	-8.16161072	-9.18181206
18	-4.046673853	-3.678794412	70	-12.24241608	-9.18181206
19	-4.133222337	-2.361841335	71	-5.1010067	-11.22221474
20	-2.206956859	-1.931087252	72	-14.28281876	-12.24241608
21	-1.366297769	-1.912816876	73	-7.14140938	-8.16161072
22	-3.096515892	-3.096515892	74	-15.3030201	-11.22221474
23	-1.227170627	-1.431699065	75	-14.28281876	-5.1010067
24	-0.704062946	-0.502902104	76	-9.18181206	-14.28281876
25	-1.306345607	-1.143052406	77	-11.22221474	-14.28281876
26	-1.78228326	-2.106334762	78	-12.24241608	-7.14140938
27	-1.831159472	-1.267725788	79	-12.24241608	-5.1010067
28	-1.124647814	-1.249608683	80	-14.28281876	-8.16161072
29	-0.973441928	-1.095122169	81	-10.2020134	-11.22221474
30	-3.892522206	-4.170559507	82	-13.26261742	-9.18181206
31	-4.254810397	-3.687502344	83	-15.3030201	-10.2020134
32	-4.254810397	-4.254810397	84	-13.26261742	-8.16161072
33	-1.946261103	-1.668223803	85	-12.24241608	-9.18181206
34	-2.959163567	-3.698954459	86	-9.18181206	-6.12120804
35	-1.749695096	-2.405830756	87	-15.3030201	-11.22221474
36	-2.557569337	-2.009518765	88	-5.1010067	-11.22221474
37	-1.180467031	-2.360934062	89	-12.24241608	-15.3030201
38	-0.986008946	-1.972017893	90	-6.12120804	-12.24241608
39	-1.905890508	-1.466069621	91	-13.26261742	-6.12120804
40	-2.085470919	-0.96252504	92	-7.14140938	-8.16161072

41	-1.102107854	-0.979651426	93	-6.12120804	-6.12120804
42	-0.848109922	-0.706758268	94	-5.1010067	-15.3030201
43	-0.86326352	-1.35655696	95	-10.2020134	-7.14140938
44	-0.941238744	-1.17654843	96	-14.28281876	-9.18181206
45	-1.303364968	-1.303364968	97	-6.12120804	-9.18181206
46	-0.770070813	-0.990091046	98	-11.22221474	-14.28281876
47	-0.724136461	-1.206894101	99	-11.22221474	-12.24241608
48	-1.760755601	-1.643371894	100	-7.14140938	-12.24241608
49	-1.057937727	-0.673233099	101	-10.2020134	-6.12120804
50	-1.004961626	-1.082266366	102	-5.1010067	-6.12120804
51	-1.108031584	-1.440441059			
52	-0.660941562	-0.849782009			
			Capex	-657.406068	-667.068406

2. Backward strategy, B-SSGS, B-PSGS

Task ID	B-SSGS	B-PSGS	Task ID	B-SSGS	B-PSGS
1	-5	-5	53	-14.28281876	-15.3030201
2	-9.320557864	-5.018761927	54	-13.26261742	-10.2020134
3	-1.652811252	-1.953322389	55	-6.12120804	-8.16161072
4	-7.294517453	-3.315689751	56	-11.22221474	-9.18181206
5	-11.76288643	-6.333861926	57	-12.24241608	-12.24241608
6	-11.69378296	-10.02324254	58	-6.12120804	-6.12120804
7	-3.687974995	-8.851139987	59	-8.16161072	-5.1010067
8	-8.837971037	-5.68155281	60	-5.1010067	-14.28281876
9	-4.507260369	-7.3242981	61	-10.2020134	-11.22221474
10	-8.741223786	-4.079237767	62	-12.24241608	-8.16161072
11	-7.575727281	-8.741223786	63	-10.2020134	-7.14140938
12	-6.719067348	-5.168513345	64	-13.26261742	-10.2020134
13	-1.466715666	-1.20004009	65	-9.18181206	-5.1010067
14	-2.979511823	-4.965853038	66	-9.18181206	-9.18181206
15	-3.816911324	-6.202480902	67	-10.2020134	-6.12120804
16	-7.597267709	-7.090783195	68	-7.14140938	-10.2020134
17	-5.392177852	-3.318263293	69	-11.22221474	-8.16161072
18	-4.782432735	-4.782432735	70	-14.28281876	-15.3030201
19	-1.476150835	-3.247531836	71	-14.28281876	-13.26261742
20	-2.206956859	-2.206956859	72	-15.3030201	-15.3030201
21	-3.005855092	-1.912816876	73	-8.16161072	-6.12120804
22	-2.858322362	-3.334709422	74	-10.2020134	-9.18181206
23	-2.454341254	-1.840755941	75	-15.3030201	-7.14140938
24	-1.508706312	-0.603482525	76	-9.18181206	-5.1010067

25	-1.306345607	-1.95951841	77	-10.2020134	-12.24241608
26	-2.106334762	-1.134180257	78	-12.24241608	-5.1010067
27	-2.112876314	-1.54944263	79	-15.3030201	-10.2020134
28	-0.874726078	-1.249608683	80	-15.3030201	-6.12120804
29	-1.825203615	-0.730081446	81	-8.16161072	-10.2020134
30	-3.336447605	-1.946261103	82	-9.18181206	-11.22221474
31	-2.552886238	-2.836540265	83	-9.18181206	-14.28281876
32	-1.418270132	-1.985578185	84	-13.26261742	-11.22221474
33	-4.170559507	-3.892522206	85	-9.18181206	-6.12120804
34	-2.219372675	-2.465969639	86	-10.2020134	-5.1010067
35	-1.968406983	-2.84325453	87	-11.22221474	-10.2020134
36	-1.278784668	-1.096101144	88	-8.16161072	-13.26261742
37	-2.360934062	-1.85501962	89	-5.1010067	-13.26261742
38	-2.112876314	-0.845150526	90	-15.3030201	-9.18181206
39	-2.199104432	-1.172855697	91	-15.3030201	-13.26261742
40	-1.283366719	-2.245891759	92	-13.26261742	-14.28281876
41	-1.102107854	-1.591933567	93	-8.16161072	-10.2020134
42	-1.837571498	-1.13081323	94	-12.24241608	-14.28281876
43	-1.849850399	-0.98658688	95	-13.26261742	-7.14140938
44	-1.647167802	-0.705929058	96	-5.1010067	-5.1010067
45	-1.203106125	-1.503882656	97	-15.3030201	-14.28281876
46	-1.540141626	-0.550050581	98	-13.26261742	-7.14140938
47	-0.563217247	-0.402298034	99	-6.12120804	-11.22221474
48	-1.408604481	-0.939069654	100	-8.16161072	-6.12120804
49	-0.96176157	-0.480880785	101	-10.2020134	-15.3030201
50	-0.541133183	-0.773047404	102	-14.28281876	-13.26261742
51	-1.551244217	-0.997228425			
52	-1.227462902	-1.416303348			
			Capex	-649.406068	-650.3360172

3. Investment measures for each scheme

[m-CCF]	SSGS	PSGS	B-SSGS	B-PSGS
NPV	**390.67**	**381.01**	**398.67**	**397.74**
NPV on equity	**441.18**	**441.18**	**441.18**	**441.18**
IRR	**21.39%**	**20.96%**	**21.77%**	**21.72%**

Appendix H: Project 2 Results

PTT Project		
Dates		
Investment date	3/1/11	
First Quarter End	3/31/11	
COD months after inv.	10	
COD Date	1/31/12	
First quarter after COD	3/31/12	
Project Life	25	years
Last quarter	3/31/37	
Financing		
Interest Rate	7.00%	pa.
Loan Term	10	year
WACC	9.16%	
Capex		
Land	30	MM Baht
EPC	900	MM Baht
Pre-investment	10	MM Baht
Contingency	3%	
	27	MM Baht
Total Costs	967	MM Baht
Life of Asset	25	year
Inverter Change	-	MM Baht
at (year)	-	
Life of Asset -		year
Exchange Rate Assumption		
1 USD =	29.96	THB
1 Euro =	40.81	THB
1 USD =	0.73	Euro
Project Cost and Financing		
Investment Cost	967.00	MM Baht
IDC	32.82	
Total Cost	999.82	
Debt	700	MM Baht
% of Debt	70%	
Equity	300	MM Baht
% of Equity	30%	
Tenor	10	years after COD
Moratorium	0	years after COD
Moratorium start	3/31/12	
Moratorium End	3/31/12	
First repayment date	6/30/12	
Last repayment date	6/30/22	

Life Time (Year)		1	2	3	4	5	6	7	8	9	10
Total Revenue		167.07	183.10	183.99	184.97	186.03	187.02	188.01	189.09	190.28	191.59
CER Revenue		-	-	-	-	-	-	-	-	-	-
Cost of Good Sold		-	-	-	-	-	-	-	-	-	-
Selling & Admin. Cost		16.86	18.59	18.91	19.24	19.58	19.92	20.27	20.64	21.01	21.40
EBITDA		150.20	164.51	165.08	165.73	166.45	167.10	167.73	168.45	169.27	170.20
Depreciation		(35.52)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)
EBIT		114.682	125.761	126.332	126.978	127.702	128.345	128.978	129.698	130.518	131.444
Net income after TAX		114.68	125.76	126.33	126.98	127.70	128.35	128.98	112.35	112.32	112.37
Add back depreciation		35.52	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75
Net Cash Flow	**-999.82**	**150.20**	**164.51**	**165.08**	**165.73**	**166.45**	**167.10**	**167.73**	**151.11**	**151.07**	**151.12**
Interest Repayment		(48.20)	(43.47)	(38.57)	(33.67)	(28.85)	(23.87)	(18.97)	(14.07)	(9.20)	(4.27)
Principal Repayment		(52.49)	(69.99)	(69.99)	(69.99)	(69.99)	(69.99)	(69.99)	(69.99)	(69.99)	(69.99)
Cash flow Available Before Dividends Payment		**49.52**	**51.06**	**56.53**	**62.07**	**67.61**	**73.24**	**78.77**	**67.05**	**71.88**	**76.86**
Discount Factor		1.0916	1.1916	1.3007	1.4199	1.5499	1.6919	1.8469	2.0161	2.2008	2.4023
PV of CF	**-999.82**	**137.60**	**138.06**	**126.92**	**116.72**	**107.39**	**98.76**	**90.82**	**74.95**	**68.65**	**62.91**
Cumulative CF		**(849.61)**	**(685.10)**	**(520.01)**	**(354.29)**	**(187.83)**	**(20.73)**	**147.00**	**298.10**	**449.18**	**600.30**

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
68.50	70.40	72.44	74.62	76.95	79.09	81.33	83.74	86.23	88.89	91.73	94.67	97.79	101.13	104.58
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18.06	18.48	18.91	19.36	19.82	20.28	20.76	21.25	21.75	22.27	22.80	23.35	23.91	24.49	25.09
50.44	51.93	53.53	55.26	57.14	58.81	60.58	62.49	64.48	66.62	68.93	71.32	73.88	76.64	79.49
(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)	(38.75)
11.693	13.173	14.777	16.511	18.385	20.054	21.827	23.742	25.733	27.871	30.175	32.568	35.132	37.884	40.742
9.98	11.20	10.34	11.56	12.87	14.04	15.28	16.62	18.01	19.51	21.12	22.80	24.59	26.52	28.52
38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75	38.75
48.74	**49.95**	**49.10**	**50.31**	**51.62**	**52.79**	**54.03**	**55.37**	**56.76**	**58.26**	**59.87**	**61.55**	**63.34**	**65.27**	**67.27**
(0.30)	(0.00)	(0.00)	-	-	-	-	-	-	-	-	-	-	-	-
(17.50)	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30.94	**49.95**	**49.10**	**50.31**	**51.62**	**52.79**	**54.03**	**55.37**	**56.76**	**58.26**	**59.87**	**61.55**	**63.34**	**65.27**	**67.27**
2.6224	2.8626	3.1248	3.4111	3.7235	4.0646	4.4369	4.8433	5.2870	5.7713	6.2999	6.8770	7.5069	8.1945	8.9452
18.58	**17.45**	**15.71**	**14.75**	**13.86**	**12.99**	**12.18**	**11.43**	**10.74**	**10.10**	**9.50**	**8.95**	**8.44**	**7.97**	**7.52**
649.03	**698.98**	**748.08**	**798.39**	**850.01**	**902.80**	**956.83**	**1,012.20**	**1,068.97**	**1,127.23**	**1,187.10**	**1,248.65**	**1,312.00**	**1,377.27**	**1,444.54**

Investment Measures	
NPV	**203.12**
NPV on Equity	**276.29**
IRR	**12.47%**

1. Forward Strategy

Task ID	SSGS	PSGS	Task ID	SSGS	PSGS
1	-20	-20	151	-11.22221474	-7.14140938
2	-8.603591874	-9.320557864	152	-5.1010067	-8.16161072
3	-1.352300115	-2.253833526	153	-7.14140938	-6.12120804
4	-3.315689751	-7.957655403	154	-13.26261742	-11.22221474
5	-10.85804902	-7.238699344	155	-10.2020134	-14.28281876
6	-5.011621268	-9.187972326	156	-11.22221474	-5.1010067
7	-5.163164993	-3.687974995	157	-14.28281876	-12.24241608
8	-6.312836455	-6.312836455	158	-8.16161072	-14.28281876
9	-4.507260369	-8.451113192	159	-10.2020134	-7.14140938
10	-4.661986019	-3.496489514	160	-9.18181206	-10.2020134
11	-5.244734271	-5.827482524	161	-8.16161072	-9.18181206
12	-5.685364679	-2.584256672	162	-5.1010067	-14.28281876
13	-0.800026727	-0.666688939	163	-15.3030201	-5.1010067
14	-4.469267734	-4.965853038	164	-12.24241608	-9.18181206
15	-3.816911324	-4.29402524	165	-6.12120804	-9.18181206
16	-3.545391598	-6.584298682	166	-9.18181206	-13.26261742
17	-6.221743675	-5.806960764	167	-8.16161072	-7.14140938
18	-5.150312176	-4.046673853	168	-8.16161072	-14.28281876
19	-2.657071502	-1.771381002	169	-8.16161072	-15.3030201
20	-4.138044111	-3.034565682	170	-9.18181206	-11.22221474
21	-3.825633753	-1.639557323	171	-8.16161072	-9.18181206
22	-3.096515892	-1.905548241	172	-9.18181206	-13.26261742
23	-3.067926568	-2.454341254	173	-14.28281876	-7.14140938
24	-0.804643367	-0.804643367	174	-15.3030201	-14.28281876
25	-0.816466004	-2.286104811	175	-10.2020134	-14.28281876
26	-0.810128755	-1.134180257	176	-6.12120804	-5.1010067
27	-1.54944263	-1.126867367	177	-13.26261742	-11.22221474
28	-1.124647814	-1.374569551	178	-12.24241608	-13.26261742
29	-0.973441928	-0.851761687	179	-11.22221474	-15.3030201
30	-2.224298404	-3.892522206	180	-12.24241608	-13.26261742
31	-1.418270132	-4.254810397	181	-5.1010067	-14.28281876
32	-2.552886238	-1.418270132	182	-15.3030201	-6.12120804
33	-2.224298404	-4.170559507	183	-13.26261742	-12.24241608
34	-2.465969639	-1.726178748	184	-13.26261742	-15.3030201
35	-1.093559435	-1.530983209	185	-8.16161072	-10.2020134
36	-2.192202289	-1.644151716	186	-8.16161072	-15.3030201
37	-2.529572209	-1.180467031	187	-6.12120804	-8.16161072
38	-1.690301051	-1.831159472	188	-10.2020134	-5.1010067
39	-1.612676583	-1.759283546	189	-10.2020134	-12.24241608
40	-1.122945879	-1.443787559	190	-11.22221474	-7.14140938
41	-0.857194998	-1.469477139	191	-5.1010067	-13.26261742
42	-1.696219844	-0.989461576	192	-12.24241608	-9.18181206
43	-1.2332336	-0.86326352	193	-15.3030201	-13.26261742

44	-0.823583901	-1.411858116	194	-12.24241608	-7.14140938
45	-1.203106125	-1.002588437	195	-13.26261742	-7.14140938
46	-0.770070813	-0.550050581	196	-10.2020134	-7.14140938
47	-0.48275764	-0.48275764	197	-9.18181206	-15.3030201
48	-1.408604481	-1.291220774	198	-14.28281876	-5.1010067
49	-1.250290041	-0.865585413	199	-14.28281876	-10.2020134
50	-0.541133183	-0.463828443	200	-7.14140938	-13.26261742
51	-1.3296379	-1.440441059	201	-11.22221474	-9.18181206
52	-1.321883125	-1.321883125	202	-5.1010067	-15.3030201
53	-5.1010067	-5.1010067	203	-12.24241608	-7.14140938
54	-9.18181206	-10.2020134	204	-13.26261742	-14.28281876
55	-7.14140938	-8.16161072	205	-14.28281876	-6.12120804
56	-10.2020134	-9.18181206	206	-14.28281876	-13.26261742
57	-12.24241608	-9.18181206	207	-11.22221474	-10.2020134
58	-10.2020134	-14.28281876	208	-13.26261742	-11.22221474
59	-7.14140938	-11.22221474	209	-14.28281876	-6.12120804
60	-12.24241608	-5.1010067	210	-7.14140938	-8.16161072
61	-8.16161072	-6.12120804	211	-8.16161072	-10.2020134
62	-14.28281876	-8.16161072	212	-14.28281876	-5.1010067
63	-7.14140938	-13.26261742	213	-15.3030201	-8.16161072
64	-15.3030201	-11.22221474	214	-6.12120804	-5.1010067
65	-13.26261742	-6.12120804	215	-15.3030201	-8.16161072
66	-11.22221474	-5.1010067	216	-7.14140938	-8.16161072
67	-10.2020134	-7.14140938	217	-15.3030201	-12.24241608
68	-13.26261742	-9.18181206	218	-13.26261742	-12.24241608
69	-11.22221474	-9.18181206	219	-11.22221474	-8.16161072
70	-15.3030201	-9.18181206	220	-13.26261742	-13.26261742
71	-8.16161072	-10.2020134	221	-15.3030201	-5.1010067
72	-14.28281876	-12.24241608	222	-14.28281876	-10.2020134
73	-9.18181206	-5.1010067	223	-8.16161072	-5.1010067
74	-8.16161072	-5.1010067	224	-7.14140938	-13.26261742
75	-7.14140938	-9.18181206	225	-5.1010067	-11.22221474
76	-6.12120804	-15.3030201	226	-15.3030201	-9.18181206
77	-10.2020134	-6.12120804	227	-10.2020134	-13.26261742
78	-6.12120804	-14.28281876	228	-11.22221474	-8.16161072
79	-10.2020134	-5.1010067	229	-12.24241608	-14.28281876
80	-9.18181206	-9.18181206	230	-12.24241608	-13.26261742
81	-12.24241608	-10.2020134	231	-12.24241608	-6.12120804
82	-12.24241608	-14.28281876	232	-10.2020134	-7.14140938
83	-10.2020134	-14.28281876	233	-13.26261742	-7.14140938
84	-8.16161072	-12.24241608	234	-9.18181206	-15.3030201
85	-8.16161072	-9.18181206	235	-13.26261742	-14.28281876
86	-6.12120804	-8.16161072	236	-9.18181206	-6.12120804
87	-9.18181206	-8.16161072	237	-7.14140938	-6.12120804
88	-13.26261742	-10.2020134	238	-6.12120804	-12.24241608
89	-15.3030201	-7.14140938	239	-11.22221474	-15.3030201

90	-11.22221474	-11.22221474	240	-12.24241608	-14.28281876
91	-8.16161072	-8.16161072	241	-9.18181206	-5.1010067
92	-9.18181206	-6.12120804	242	-6.12120804	-11.22221474
93	-15.3030201	-14.28281876	243	-11.22221474	-14.28281876
94	-13.26261742	-7.14140938	244	-10.2020134	-8.16161072
95	-9.18181206	-7.14140938	245	-7.14140938	-8.16161072
96	-9.18181206	-9.18181206	246	-15.3030201	-9.18181206
97	-12.24241608	-10.2020134	247	-13.26261742	-14.28281876
98	-11.22221474	-6.12120804	248	-5.1010067	-8.16161072
99	-13.26261742	-8.16161072	249	-10.2020134	-6.12120804
100	-15.3030201	-6.12120804	250	-14.28281876	-15.3030201
101	-6.12120804	-9.18181206	251	-14.28281876	-8.16161072
102	-15.3030201	-14.28281876	252	-9.18181206	-7.14140938
103	-13.26261742	-10.2020134	253	-11.22221474	-6.12120804
104	-7.14140938	-6.12120804	254	-10.2020134	-7.14140938
105	-15.3030201	-15.3030201	255	-5.1010067	-6.12120804
106	-14.28281876	-7.14140938	256	-14.28281876	-7.14140938
107	-15.3030201	-11.22221474	257	-13.26261742	-5.1010067
108	-13.26261742	-12.24241608	258	-5.1010067	-10.2020134
109	-10.2020134	-14.28281876	259	-5.1010067	-7.14140938
110	-15.3030201	-12.24241608	260	-9.18181206	-6.12120804
111	-9.18181206	-10.2020134	261	-9.18181206	-15.3030201
112	-10.2020134	-12.24241608	262	-13.26261742	-15.3030201
113	-15.3030201	-11.22221474	263	-12.24241608	-6.12120804
114	-8.16161072	-7.14140938	264	-9.18181206	-15.3030201
115	-11.22221474	-10.2020134	265	-5.1010067	-11.22221474
116	-7.14140938	-15.3030201	266	-13.26261742	-10.2020134
117	-15.3030201	-10.2020134	267	-11.22221474	-8.16161072
118	-13.26261742	-7.14140938	268	-7.14140938	-6.12120804
119	-9.18181206	-9.18181206	269	-11.22221474	-9.18181206
120	-11.22221474	-11.22221474	270	-9.18181206	-6.12120804
121	-6.12120804	-15.3030201	271	-12.24241608	-9.18181206
122	-6.12120804	-15.3030201	272	-7.14140938	-9.18181206
123	-14.28281876	-9.18181206	273	-7.14140938	-12.24241608
124	-12.24241608	-9.18181206	274	-5.1010067	-6.12120804
125	-5.1010067	-10.2020134	275	-9.18181206	-5.1010067
126	-9.18181206	-13.26261742	276	-14.28281876	-6.12120804
127	-14.28281876	-14.28281876	277	-8.16161072	-7.14140938
128	-5.1010067	-13.26261742	278	-10.2020134	-8.16161072
129	-10.2020134	-15.3030201	279	-11.22221474	-11.22221474
130	-12.24241608	-13.26261742	280	-14.28281876	-13.26261742
131	-6.12120804	-6.12120804	281	-12.24241608	-8.16161072
132	-10.2020134	-11.22221474	282	-12.24241608	-10.2020134
133	-9.18181206	-15.3030201	283	-9.18181206	-5.1010067
134	-19.73434705	-14.09596218	284	-15.3030201	-15.3030201
135	-12.24241608	-8.16161072	285	-8.16161072	-9.18181206

136	-9.18181206	-11.22221474	286	-5.1010067	-6.12120804
137	-9.18181206	-11.22221474	287	-9.18181206	-9.18181206
138	-14.28281876	-13.26261742	288	-14.28281876	-6.12120804
139	-13.26261742	-6.12120804	289	-8.16161072	-7.14140938
140	-13.26261742	-11.22221474	290	-10.2020134	-7.14140938
141	-7.14140938	-15.3030201	291	-14.28281876	-7.14140938
142	-11.22221474	-7.14140938	292	-10.2020134	-15.3030201
143	-10.2020134	-6.12120804	293	-15.3030201	-6.12120804
144	-13.26261742	-14.28281876	294	-10.2020134	-15.3030201
145	-12.24241608	-9.18181206	295	-11.22221474	-9.18181206
146	-10.2020134	-11.22221474	296	-14.28281876	-5.1010067
147	-8.16161072	-9.18181206	297	-13.26261742	-6.12120804
148	-14.28281876	-8.16161072	298	-5.1010067	-7.14140938
149	-15.3030201	-9.18181206			
150	-11.22221474	-6.12120804			
			Capex	-1009.5789	-1012.446

2. Backward strategy

Task ID	B-SSGS	B-PSGS	Task ID	B-SSGS	B-PSGS
1	-19.45735837	-19.45735837	151	-11.22221474	-19.38382546
2	-10.75448984	-10.75448984	152	-7.14140938	-17.34342278
3	-1.652811252	-3.005111367	153	-18.36362412	-16.32322144
4	-12.59962106	-3.978827702	154	-20.4040268	-9.18181206
5	-9.04837418	-5.429024508	155	-14.28281876	-20.4040268
6	-8.352702114	-10.85851275	156	-17.34342278	-14.28281876
7	-6.63835499	-8.851139987	157	-14.28281876	-8.16161072
8	-8.837971037	-10.10053833	158	-16.32322144	-9.18181206
9	-9.577928284	-2.817037731	159	-15.3030201	-17.34342278
10	-4.079237767	-9.90672029	160	-7.14140938	-14.28281876
11	-4.661986019	-2.913741262	161	-7.14140938	-5.1010067
12	-5.168513345	-5.685364679	162	-14.28281876	-6.12120804
13	-0.933364515	-0.800026727	163	-10.2020134	-15.3030201
14	-5.959023645	-3.476097127	164	-12.24241608	-14.28281876
15	-2.385569578	-6.202480902	165	-15.3030201	-19.38382546
16	-5.571329654	-5.571329654	166	-18.36362412	-10.2020134
17	-3.733046205	-3.318263293	167	-11.22221474	-11.22221474
18	-4.046673853	-4.782432735	168	-14.28281876	-16.32322144
19	-2.952301669	-5.314143005	169	-10.2020134	-5.1010067
20	-3.586304896	-1.655217645	170	-8.16161072	-19.38382546
21	-2.459335984	-2.18607643	171	-8.16161072	-14.28281876
22	-3.334709422	-3.811096482	172	-13.26261742	-16.32322144
23	-2.658869692	-3.272455005	173	-19.38382546	-6.12120804
24	-1.709867154	-1.911027996	174	-17.34342278	-12.24241608
25	-1.632932008	-0.816466004	175	-15.3030201	-17.34342278

26	-2.592412015	-2.916463517	176	-18.36362412	-13.26261742
27	-2.676309997	-1.972017893	177	-16.32322144	-12.24241608
28	-1.749452156	-0.624804341	178	-10.2020134	-16.32322144
29	-0.973441928	-2.068564097	179	-15.3030201	-11.22221474
30	-1.946261103	-1.390186502	180	-17.34342278	-16.32322144
31	-4.254810397	-3.687502344	181	-15.3030201	-12.24241608
32	-5.389426503	-3.120194291	182	-5.1010067	-20.4040268
33	-4.726634108	-5.004671408	183	-10.2020134	-17.34342278
34	-4.438745351	-3.945551423	184	-10.2020134	-8.16161072
35	-3.280678304	-3.499390191	185	-10.2020134	-17.34342278
36	-1.826835241	-2.009518765	186	-16.32322144	-6.12120804
37	-2.192295914	-1.011828884	187	-15.3030201	-12.24241608
38	-1.972017893	-2.394593156	188	-19.38382546	-17.34342278
39	-1.172855697	-1.466069621	189	-20.4040268	-20.4040268
40	-0.8021042	-1.443787559	190	-19.38382546	-10.2020134
41	-0.612282141	-0.857194998	191	-14.28281876	-16.32322144
42	-0.706758268	-1.978923152	192	-6.12120804	-6.12120804
43	-1.35655696	-2.096497119	193	-16.32322144	-10.2020134
44	-1.764822645	-1.17654843	194	-20.4040268	-14.28281876
45	-0.902329594	-2.005176874	195	-8.16161072	-5.1010067
46	-1.650151743	-1.650151743	196	-12.24241608	-20.4040268
47	-1.367813315	-1.287353708	197	-8.16161072	-5.1010067
48	-1.291220774	-2.112906721	198	-10.2020134	-13.26261742
49	-1.442642355	-1.442642355	199	-10.2020134	-7.14140938
50	-1.082266366	-0.618437924	200	-6.12120804	-10.2020134
51	-1.772850534	-0.554015792	201	-9.18181206	-5.1010067
52	-1.416303348	-0.566521339	202	-14.28281876	-13.26261742
53	-7.14140938	-11.22221474	203	-18.36362412	-7.14140938
54	-16.32322144	-13.26261742	204	-20.4040268	-7.14140938
55	-10.2020134	-8.16161072	205	-19.38382546	-15.3030201
56	-5.1010067	-16.32322144	206	-16.32322144	-20.4040268
57	-16.32322144	-17.34342278	207	-14.28281876	-13.26261742
58	-20.4040268	-14.28281876	208	-12.24241608	-19.38382546
59	-19.38382546	-19.38382546	209	-9.18181206	-11.22221474
60	-12.24241608	-5.1010067	210	-20.4040268	-9.18181206
61	-11.22221474	-14.28281876	211	-11.22221474	-20.4040268
62	-16.32322144	-15.3030201	212	-13.26261742	-10.2020134
63	-16.32322144	-19.38382546	213	-18.36362412	-12.24241608
64	-17.34342278	-17.34342278	214	-16.32322144	-9.18181206
65	-15.3030201	-11.22221474	215	-8.16161072	-16.32322144
66	-16.32322144	-15.3030201	216	-14.28281876	-6.12120804
67	-5.1010067	-18.36362412	217	-11.22221474	-5.1010067
68	-16.32322144	-18.36362412	218	-15.3030201	-17.34342278
69	-18.36362412	-8.16161072	219	-8.16161072	-5.1010067
70	-11.22221474	-10.2020134	220	-14.28281876	-13.26261742
71	-19.38382546	-7.14140938	221	-11.22221474	-15.3030201

72	-20.4040268	-15.3030201	222	-6.12120804	-17.34342278
73	-14.28281876	-14.28281876	223	-9.18181206	-7.14140938
74	-15.3030201	-13.26261742	224	-19.38382546	-8.16161072
75	-15.3030201	-7.14140938	225	-13.26261742	-6.12120804
76	-17.34342278	-11.22221474	226	-7.14140938	-18.36362412
77	-10.2020134	-7.14140938	227	-9.18181206	-9.18181206
78	-15.3030201	-14.28281876	228	-15.3030201	-6.12120804
79	-5.1010067	-16.32322144	229	-5.1010067	-13.26261742
80	-12.24241608	-9.18181206	230	-8.16161072	-18.36362412
81	-7.14140938	-14.28281876	231	-11.22221474	-9.18181206
82	-9.18181206	-20.4040268	232	-12.24241608	-14.28281876
83	-13.26261742	-18.36362412	233	-16.32322144	-15.3030201
84	-7.14140938	-9.18181206	234	-18.36362412	-16.32322144
85	-12.24241608	-10.2020134	235	-9.18181206	-10.2020134
86	-17.34342278	-19.38382546	236	-10.2020134	-16.32322144
87	-9.18181206	-18.36362412	237	-14.28281876	-17.34342278
88	-9.18181206	-14.28281876	238	-13.26261742	-8.16161072
89	-12.24241608	-13.26261742	239	-7.14140938	-20.4040268
90	-7.14140938	-11.22221474	240	-11.22221474	-10.2020134
91	-20.4040268	-18.36362412	241	-10.2020134	-11.22221474
92	-6.12120804	-8.16161072	242	-7.14140938	-16.32322144
93	-12.24241608	-7.14140938	243	-15.3030201	-18.36362412
94	-11.22221474	-17.34342278	244	-13.26261742	-15.3030201
95	-8.16161072	-17.34342278	245	-15.3030201	-17.34342278
96	-11.22221474	-5.1010067	246	-19.38382546	-13.26261742
97	-18.36362412	-13.26261742	247	-11.22221474	-10.2020134
98	-14.28281876	-15.3030201	248	-13.26261742	-20.4040268
99	-15.3030201	-13.26261742	249	-16.32322144	-6.12120804
100	-19.38382546	-16.32322144	250	-5.1010067	-8.16161072
101	-13.26261742	-12.24241608	251	-14.28281876	-15.3030201
102	-12.24241608	-5.1010067	252	-12.24241608	-11.22221474
103	-10.2020134	-6.12120804	253	-10.2020134	-6.12120804
104	-5.1010067	-7.14140938	254	-11.22221474	-7.14140938
105	-6.12120804	-10.2020134	255	-20.4040268	-17.34342278
106	-18.36362412	-7.14140938	256	-15.3030201	-5.1010067
107	-18.36362412	-10.2020134	257	-20.4040268	-14.28281876
108	-17.34342278	-6.12120804	258	-19.38382546	-11.22221474
109	-18.36362412	-19.38382546	259	-19.38382546	-12.24241608
110	-7.14140938	-15.3030201	260	-15.3030201	-13.26261742
111	-19.38382546	-13.26261742	261	-12.24241608	-6.12120804
112	-16.32322144	-8.16161072	262	-20.4040268	-9.18181206
113	-17.34342278	-11.22221474	263	-20.4040268	-5.1010067
114	-6.12120804	-9.18181206	264	-12.24241608	-14.28281876
115	-18.36362412	-7.14140938	265	-10.2020134	-9.18181206
116	-7.14140938	-13.26261742	266	-7.14140938	-5.1010067
117	-8.16161072	-8.16161072	267	-18.36362412	-9.18181206

118	-18.36362412	-7.14140938	268	-20.4040268	-13.26261742
119	-14.28281876	-18.36362412	269	-7.14140938	-20.4040268
120	-10.2020134	-20.4040268	270	-8.16161072	-13.26261742
121	-15.3030201	-17.34342278	271	-17.34342278	-20.4040268
122	-16.32322144	-18.36362412	272	-17.34342278	-19.38382546
123	-13.26261742	-7.14140938	273	-10.2020134	-9.18181206
124	-7.14140938	-14.28281876	274	-7.14140938	-16.32322144
125	-8.16161072	-13.26261742	275	-12.24241608	-14.28281876
126	-6.12120804	-10.2020134	276	-9.18181206	-15.3030201
127	-15.3030201	-13.26261742	277	-8.16161072	-20.4040268
128	-13.26261742	-15.3030201	278	-5.1010067	-12.24241608
129	-6.12120804	-10.2020134	279	-7.14140938	-7.14140938
130	-17.34342278	-10.2020134	280	-10.2020134	-9.18181206
131	-5.1010067	-20.4040268	281	-19.38382546	-18.36362412
132	-10.2020134	-18.36362412	282	-19.38382546	-12.24241608
133	-9.18181206	-8.16161072	283	-13.26261742	-16.32322144
134	-39.4686941	-14.09596218	284	-8.16161072	-7.14140938
135	-10.2020134	-6.12120804	285	-12.24241608	-16.32322144
136	-17.34342278	-19.38382546	286	-14.28281876	-15.3030201
137	-16.32322144	-14.28281876	287	-14.28281876	-19.38382546
138	-5.1010067	-18.36362412	288	-16.32322144	-10.2020134
139	-15.3030201	-7.14140938	289	-20.4040268	-5.1010067
140	-18.36362412	-20.4040268	290	-17.34342278	-18.36362412
141	-8.16161072	-7.14140938	291	-10.2020134	-12.24241608
142	-9.18181206	-16.32322144	292	-15.3030201	-9.18181206
143	-16.32322144	-13.26261742	293	-5.1010067	-14.28281876
144	-12.24241608	-7.14140938	294	-9.18181206	-15.3030201
145	-17.34342278	-17.34342278	295	-10.2020134	-10.2020134
146	-13.26261742	-5.1010067	296	-12.24241608	-12.4040268
147	-19.38382546	-19.38382546	297	-11.22221474	-12.24241608
148	-18.36362412	-9.18181206	298	-6.12120804	-9.18181206
149	-20.4040268	-9.18181206			
150	-16.32322144	-16.32322144			
			Capex	-932.2125848	-988.9754644

3. Investment measures for each scheme

[m-CCF]	SSGS	PSGS	B-SSGS	B-PSGS
NPV	**193.36**	**190.49**	**270.72**	**213.96**
NPV on equity	**276.29**	**276.29**	**276.29**	**276.29**
IRR	**12.29%**	**12.23%**	**13.86%**	**12.68%**

Appendix I: Project 3 Results

PTT Project		
Dates		
Investment date	3/1/11	
First Quarter End	3/31/11	
COD months after inv.	10	
COD Date	1/31/12	
First quarter after COD	3/31/12	
Project Life	10	years
Last quarter	3/31/22	
Financing		
Interest Rate	7.00%	pa.
Loan Term	10	year
WACC	9.16%	
Capex		
Land	15	MM Baht
EPC	600	MM Baht
Pre-investment	5	MM Baht
Contingency	3%	
	18	MM Baht
Total Costs	638	MM Baht
Life of Asset	10	year
Inverter Change	-	MM Baht
at (year)	-	
Life of Asset -		year
Exchange Rate Assumption		
1 USD =	29.96	THB
1 Euro =	40.81	THB
1 USD =	0.73	Euro
Project Cost and Financing		
Investment Cost	638.00	MM Baht
IDC	21.65	
Total Cost	659.65	
Debt	462	MM Baht
% of Debt	70%	
Equity	198	MM Baht
% of Equity	30%	
Tenor	10	years after COD
Moratorium	0	years after COD
Moratorium start	3/31/12	
Moratorium End	3/31/12	
First repayment date	6/30/12	
Last repayment date	6/30/22	

Life Time (Year)		1	2	3	4	5	6	7	8	9	10
Total Revenue		167.07	183.10	183.99	184.97	186.03	187.02	188.01	189.09	190.28	191.59
CER Revenue		-	-	-	-	-	-	-	-	-	-
Cost of Good Sold		-	-	-	-	-	-	-	-	-	-
Selling & Admin. Cost		13.74	15.40	15.66	15.93	16.20	16.48	16.76	17.06	17.36	17.67
EBITDA		153.32	167.69	168.33	169.04	169.83	170.54	171.24	172.03	172.92	173.92
Depreciation		(59.05)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)	(64.41)
EBIT		94.277	103.279	103.914	104.624	105.415	106.126	106.828	107.617	108.509	109.509
Tax		-	-	-	-	-	-	-	-14.75	-15.37	-16.00
Net income after TAX		94.28	103.28	103.91	104.62	105.42	106.13	106.83	92.87	93.14	93.51
Add back depreciation		59.05	64.41	64.41	64.41	64.41	64.41	64.41	64.41	64.41	64.41
Net Cash Flow	**-659.65**	**153.32**	**167.69**	**168.33**	**169.04**	**169.83**	**170.54**	**171.24**	**157.28**	**157.56**	**157.92**
Interest Repayment		(31.80)	(28.68)	(25.45)	(22.21)	(19.04)	(15.75)	(12.52)	(9.29)	(6.07)	(2.82)
Principal Repayment		(34.63)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)	(46.18)
Cash flow Available Before Dividends Payment		**86.89**	**92.84**	**96.71**	**100.65**	**104.62**	**108.61**	**112.55**	**101.82**	**105.31**	**108.92**
Discount Factor		1.0916	1.1916	1.3007	1.4199	1.5499	1.6919	1.8469	2.0161	2.2008	2.4023
PV of CF	**-659.65**	**140.46**	**140.73**	**129.41**	**119.05**	**109.57**	**100.80**	**92.72**	**78.01**	**71.59**	**65.74**
Cumulative CF		13.74	15.40	15.66	15.93	16.20	16.48	16.76	17.06	17.36	17.67

Investment Measures	
NPV	**388.42**
NPV on Equity	**441.18**
IRR	**21.29%**

1. Forward strategy

Task ID	SSGS	PSGS	Task ID	SSGS	PSGS
1	-5	-5	53	-13.26261742	-11.22221474
2	-9.320557864	-8.603591874	54	-6.12120804	-12.24241608
3	-1.953322389	-1.502555684	55	-15.3030201	-14.28281876
4	-9.283931304	-3.315689751	56	-10.2020134	-15.3030201
5	-8.143536762	-9.953211598	57	-7.14140938	-9.18181206
6	-11.69378296	-8.352702114	58	-6.12120804	-11.22221474
7	-3.687974995	-4.425569994	59	-11.22221474	-5.1010067
8	-9.469254683	-8.206687392	60	-11.22221474	-15.3030201
9	-6.760890554	-6.760890554	61	-14.28281876	-7.14140938
10	-2.913741262	-2.913741262	62	-7.14140938	-6.12120804
11	-3.496489514	-6.410230776	63	-8.16161072	-6.12120804
12	-3.617959341	-7.235918683	64	-12.24241608	-7.14140938
13	-2.000066817	-1.733391242	65	-7.14140938	-9.18181206
14	-5.959023645	-3.476097127	66	-13.26261742	-13.26261742
15	-2.862683493	-5.725366986	67	-5.1010067	-11.22221474
16	-3.038907084	-7.597267709	68	-14.28281876	-9.18181206
17	-5.806960764	-4.147829117	69	-8.16161072	-9.18181206
18	-4.046673853	-3.678794412	70	-12.24241608	-9.18181206
19	-4.133222337	-2.361841335	71	-5.1010067	-11.22221474
20	-2.206956859	-1.931087252	72	-14.28281876	-12.24241608
21	-1.366297769	-1.912816876	73	-7.14140938	-8.16161072
22	-3.096515892	-3.096515892	74	-15.3030201	-11.22221474
23	-1.227170627	-1.431699065	75	-14.28281876	-5.1010067
24	-0.704062946	-0.502902104	76	-9.18181206	-14.28281876
25	-1.306345607	-1.143052406	77	-11.22221474	-14.28281876
26	-1.78228326	-2.106334762	78	-12.24241608	-7.14140938
27	-1.831159472	-1.267725788	79	-12.24241608	-5.1010067
28	-1.124647814	-1.249608683	80	-14.28281876	-8.16161072
29	-0.973441928	-1.095122169	81	-10.2020134	-11.22221474
30	-3.892522206	-4.170559507	82	-13.26261742	-9.18181206
31	-4.254810397	-3.687502344	83	-15.3030201	-10.2020134
32	-4.254810397	-4.254810397	84	-13.26261742	-8.16161072
33	-1.946261103	-1.668223803	85	-12.24241608	-9.18181206
34	-2.959163567	-3.698954459	86	-9.18181206	-6.12120804
35	-1.749695096	-2.405830756	87	-15.3030201	-11.22221474
36	-2.557569337	-2.009518765	88	-5.1010067	-11.22221474
37	-1.180467031	-2.360934062	89	-12.24241608	-15.3030201
38	-0.986008946	-1.972017893	90	-6.12120804	-12.24241608
39	-1.905890508	-1.466069621	91	-13.26261742	-6.12120804
40	-2.085470919	-0.96252504	92	-7.14140938	-8.16161072
41	-1.102107854	-0.979651426	93	-6.12120804	-6.12120804

42	-0.848109922	-0.706758268	94	-5.1010067	-15.3030201
43	-0.86326352	-1.35655696	95	-10.2020134	-7.14140938
44	-0.941238744	-1.17654843	96	-14.28281876	-9.18181206
45	-1.303364968	-1.303364968	97	-6.12120804	-9.18181206
46	-0.770070813	-0.990091046	98	-11.22221474	-14.28281876
47	-0.724136461	-1.206894101	99	-11.22221474	-12.24241608
48	-1.760755601	-1.643371894	100	-7.14140938	-12.24241608
49	-1.057937727	-0.673233099	101	-10.2020134	-6.12120804
50	-1.004961626	-1.082266366	102	-5.1010067	-6.12120804
51	-1.108031584	-1.440441059			
52	-0.660941562	-0.849782009			
			Capex	-689.406068	-686.9489934

2. Backward strategy

Task ID	B-SSGS	B-PSGS	Task ID	B-SSGS	B-PSGS
1	-5	-5	53	-13.26261742	-11.22221474
2	-9.320557864	-8.603591874	54	-6.12120804	-12.24241608
3	-1.953322389	-1.502555684	55	-15.3030201	-14.28281876
4	-9.283931304	-3.315689751	56	-10.2020134	-15.3030201
5	-8.143536762	-9.953211598	57	-7.14140938	-9.18181206
6	-11.69378296	-8.352702114	58	-6.12120804	-11.22221474
7	-3.687974995	-4.425569994	59	-11.22221474	-5.1010067
8	-9.469254683	-8.206687392	60	-11.22221474	-15.3030201
9	-6.760890554	-6.760890554	61	-14.28281876	-7.14140938
10	-2.913741262	-2.913741262	62	-7.14140938	-6.12120804
11	-3.496489514	-6.410230776	63	-8.16161072	-6.12120804
12	-3.617959341	-7.235918683	64	-12.24241608	-7.14140938
13	-2.000066817	-1.733391242	65	-7.14140938	-9.18181206
14	-5.959023645	-3.476097127	66	-13.26261742	-13.26261742
15	-2.862683493	-5.725366986	67	-5.1010067	-11.22221474
16	-3.038907084	-7.597267709	68	-14.28281876	-9.18181206
17	-5.806960764	-4.147829117	69	-8.16161072	-9.18181206
18	-4.046673853	-3.678794412	70	-12.24241608	-9.18181206
19	-4.133222337	-2.361841335	71	-5.1010067	-11.22221474
20	-2.206956859	-1.931087252	72	-14.28281876	-12.24241608
21	-1.366297769	-1.912816876	73	-7.14140938	-8.16161072
22	-3.096515892	-3.096515892	74	-15.3030201	-11.22221474
23	-1.227170627	-1.431699065	75	-14.28281876	-5.1010067
24	-0.704062946	-0.502902104	76	-9.18181206	-14.28281876
25	-1.306345607	-1.143052406	77	-11.22221474	-14.28281876

26	-1.78228326	-2.106334762	78	-12.24241608	-7.14140938
27	-1.831159472	-1.267725788	79	-12.24241608	-5.1010067
28	-1.124647814	-1.249608683	80	-14.28281876	-8.16161072
29	-0.973441928	-1.095122169	81	-10.2020134	-11.22221474
30	-3.892522206	-4.170559507	82	-13.26261742	-9.18181206
31	-4.254810397	-3.687502344	83	-15.3030201	-10.2020134
32	-4.254810397	-4.254810397	84	-13.26261742	-8.16161072
33	-1.946261103	-1.668223803	85	-12.24241608	-9.18181206
34	-2.959163567	-3.698954459	86	-9.18181206	-6.12120804
35	-1.749695096	-2.405830756	87	-15.3030201	-11.22221474
36	-2.557569337	-2.009518765	88	-5.1010067	-11.22221474
37	-1.180467031	-2.360934062	89	-12.24241608	-15.3030201
38	-0.986008946	-1.972017893	90	-6.12120804	-12.24241608
39	-1.905890508	-1.466069621	91	-13.26261742	-6.12120804
40	-2.085470919	-0.96252504	92	-7.14140938	-8.16161072
41	-1.102107854	-0.979651426	93	-6.12120804	-6.12120804
42	-0.848109922	-0.706758268	94	-5.1010067	-15.3030201
43	-0.86326352	-1.35655696	95	-10.2020134	-7.14140938
44	-0.941238744	-1.17654843	96	-14.28281876	-9.18181206
45	-1.303364968	-1.303364968	97	-6.12120804	-9.18181206
46	-0.770070813	-0.990091046	98	-11.22221474	-14.28281876
47	-0.724136461	-1.206894101	99	-11.22221474	-12.24241608
48	-1.760755601	-1.643371894	100	-7.14140938	-12.24241608
49	-1.057937727	-0.673233099	101	-10.2020134	-6.12120804
50	-1.004961626	-1.082266366	102	-5.1010067	-6.12120804
51	-1.108031584	-1.440441059			
52	-0.660941562	-0.849782009			
			Capex	-656.017233	-653.3360172

3. Investment measures for each scheme

[m-CCF]	SSGS	PSGS	B-SSGS	B-PSGS
NPV	**358.67**	**361.13**	**392.06**	**394.74**
NPV on equity	**441.18**	**441.18**	**441.18**	**441.18**
IRR	**19.98%**	**20.08%**	**21.46%**	**21.58%**

Appendix J: List of Publications

Conference papers

Tantisuvanichkul, V. and Kidd, M. (2011), Project scheduling a review through literature, in proceedings of RICS Construction and Property Conference (COBRA 2011), University of Salford, Manchester, UK, 12-13 September, ISBN 978-1-907842-19-1.

Tantisuvanichkul, V. and Kidd, M. (2011), Maximizing Net Present Value a review through literature, in proceedings of 2011 2nd International Conference on Construction and Project Management (ICCPM), Singapore, 16-18 September, ISBN 978- 981-08-9176-3.

Tantisuvanichkul, V. and Kidd, M. (2011), Improve Net Present Value using cash flow weight, in proceedings of 2011 2nd International Conference on Construction and Project Management (ICCPM), Singapore, 16-18 September, ISBN 978- 981-08-9176-3.

Appendix K: m-CCF software

This study use variety of software.

The proposed priority rule based heuristic is implemented in two schemes (serial and parallel) and two strategies (forward and backward). All of the scheduling rules and schemes of the proposed technique are coded in MATLAB (R2011a). The screenshot of MATLAB (R2011a) can be seen in Figure 1.

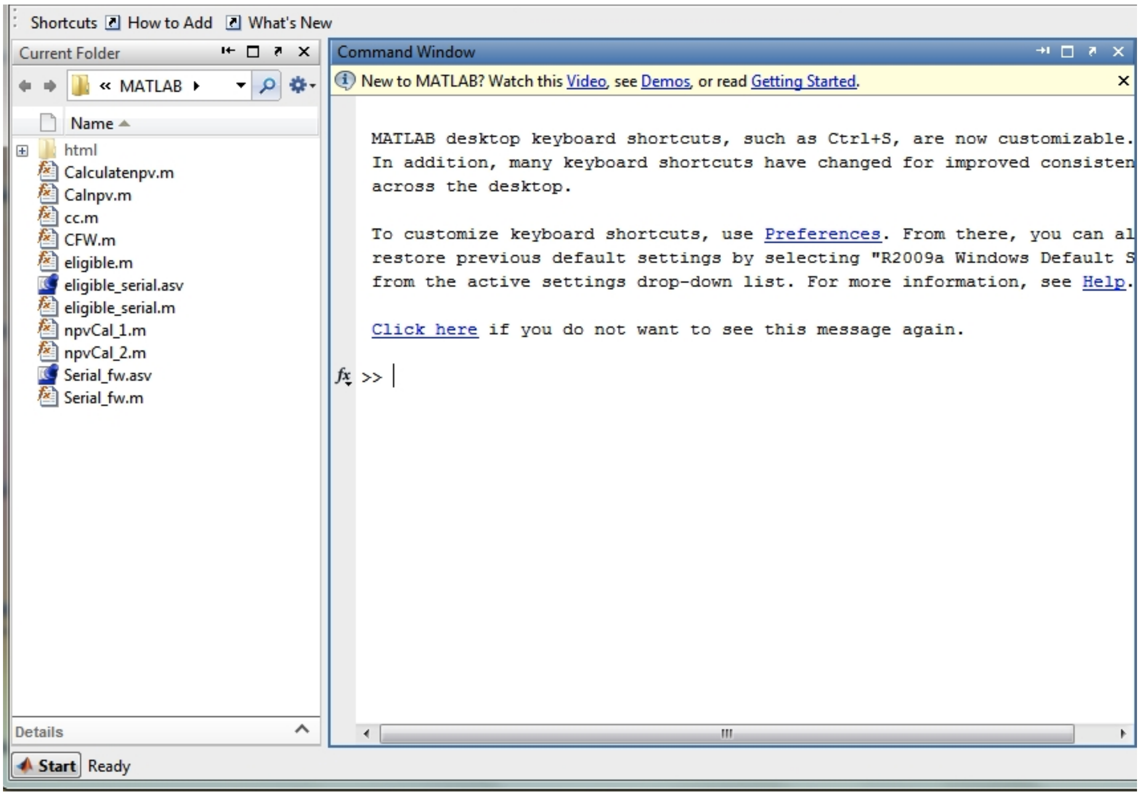


Figure 1: The screenshot of MATLAB (R2011a)

Solver tools of the latest version of Excel spreadsheet are used to analyse the research data collected and facilitate searching the optimal solution.

The solution is solved with GRG optimisation routine as can be seen in Figure 2. The Objective cell is set to maximise the NPV value subject to the constraints.

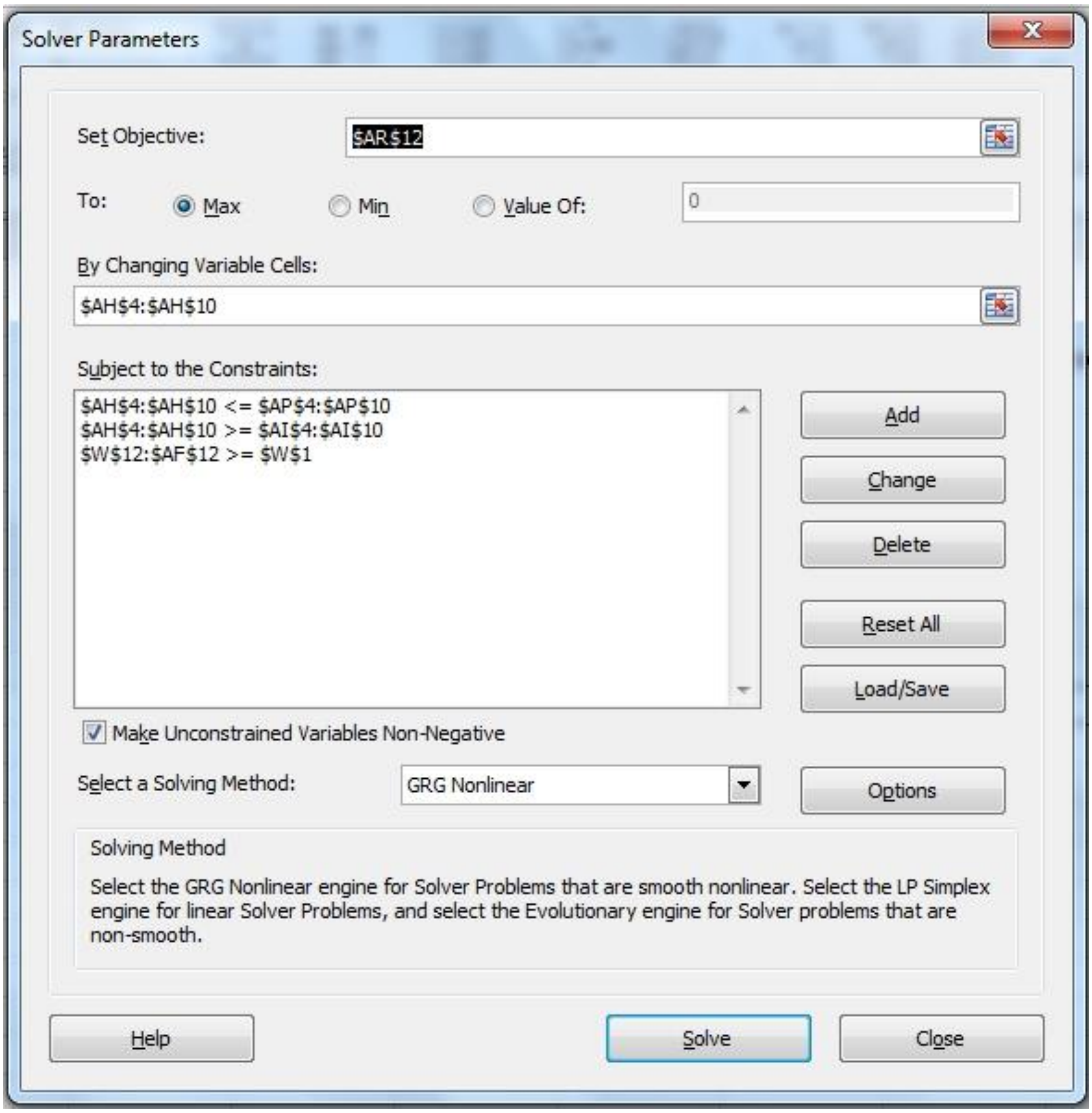


Figure 2: The screenshot of Solver tools