Fall 2009

Resource assignment in short life technology intensive (SLTI) new product development (NPD)

Randy A. Reagan
New Jersey Institute of Technology

ABSTRACT

RESOURCE ASSIGNMENT IN
SHORT LIFE TECHNOLOGY INTENSIVE (SLTI)
NEW PRODUCT DEVELOPMENT (NDP)

by

Randy A. Reagan

Enterprises managing multiple concurrent New Product Development (NPD) projects face significant challenges assigning staff to projects in order to achieve launch schedules that maximize financial returns. The challenge is increased with the class of Short Life Technology Intensive (SLTI) products characterized by technical complexity, short development cycles and short revenue life cycles. Technical complexity drives the need to assign staffing resources of various technical disciplines and skill levels. SLTI products are rapidly developed and launched into stationary market windows where the revenue life cycle is short and decreasing with any time-to-market delay. The SLTI-NPD project management decision is to assign staff of varying technical discipline and skill level to minimize the revenue loss due to product launch delays across multiple projects.

This dissertation considers an NPD organization responsible for multiple concurrent SLTI projects each characterized by a set of tasks having technical discipline requirements, task duration estimates and logical precedence relationships. Each project has a known potential launch date and potential revenue life cycle. The organization has a group of technical professionals characterized by a range of skill levels in a known set of technical disciplines.

The SLTI-NPD resource assignment problem is solved using a multi-step process referred to as the Resource Assignment and Multi-Project Scheduling (RAMPS) decision support tool. Robust scheduling techniques are integrated to develop schedules that consider variation in task and project duration estimates. A valuation function provides a time-value linkage between schedules and the product revenue life cycle for each product. Productivity metrics are developed as the basis for prioritizing projects for resources assignment.

The RAMPS tool implements assignment and scheduling algorithms in two phases; (i) a constructive approach that employs priority rule heuristics to derive feasible assignments and schedules and (ii) an improvement heuristic that considers productivity gains that can be achieved by interchanging resources of differing skill levels and corresponding work rates. An experimental analysis is conducted using the RAMPS tool and simulated project and resource data sets. Results show significant productivity and efficiency gains that can be achieved through effective project and resource prioritization and by including consideration of skill level in the assignment of technical resources.

RESOURCE ASSIGNMENT IN SHORT LIFE TECHNOLOGY INTENSIVE (SLTI) NEW PRODUCT DEVELOPMENT (NPD)

by

Randy A. Reagan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for Degree of
Doctor of Philosophy in Industrial Engineering

Department of Mechanical and Industrial Engineering

January 2010

APPROVAL PAGE

RESOURCE ASSIGNMENT IN SHORT LIFE TECHNOLOGY INTENSIVE (SLTI) NEW PRODUCT DEVELOPMENT (NPD)

Randy A. Reagan

11/2/2009

Dr. Sanchoy Das, Dissertation Advisor Date
Professor of Mechanical and Industrial Engineering, NJIT

11/2/09

Dr. Athanassios Bladikas, Committee Member Date
Associate Professor of Mechanical and Industrial Engineering, NJIT

11/2/09

Mr. Robert English, Committee Member Date
Interim Dean of the School of Management, NJIT

11/2/09

Dr. Paul Ranky, Committee Member Date
Professor of Mechanical and Industrial Engineering, NJIT

11/2/2009

Dr. Carl Wolf, Committee Member Date
Professor Emeritus of Mechanical and Industrial Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Randy A. Reagan

Degree: Doctor of Philosophy

Date: January 2010

Department: Mechanical and Industrial Engineering

Education

- Doctor of Philosophy in Industrial Engineering,
  New Jersey Institute of Technology, Newark, NJ

- Master of Business Administration, Statistics and Operations Research,
  New York University, New York, NY

- Master of Science in Mechanical Engineering,
  University of Michigan, Ann Arbor, MI

- Bachelor of Science in Mechanical Engineering, Magna cum Laude,
  New Jersey Institute of Technology, Newark, NJ

Publications and Presentations:

Reagan, R., "Time-Value Model for Prioritizing New Product Development (NPD) Projects in Stationary Market Windows," NJIT Research Forum, April 2009.

Reagan, R., Y. Lu, E. Gronvall, R. Hagen, "Miniature Outside Plant Plug-and-Play Solutions," FTTH Proceedings, September 2008.

Reagan, R., R. Hagen, "Innovative Installation Techniques for Drop Terminals," FTTH Proceedings, September 2007.

Reagan, R., "Creating a Cost-Effective Plug-and-Play Architecture," Lightwave, May 2006.

Reagan, R., "Rugged Connector Technology," Broadband Properties, September 2005.

Reagan, R., J. Gniadek, Y. Lu, "Outside Plant Connectors You Can Rely On," FTTH Proceedings, October 2005.

Reagan, R., J. Gniadek, Y. Lu, "Multiple Solutions for Multiple Dwelling Units," FTTH Proceedings, October 2005.

Reagan, R., J. Gniadek, Y. Lu, "It's Still Happening at the Hub," FTTH Proceedings, October 2005.

Reagan, R., "It's Happening at the Hub," FTTH Proceedings, October 2004.

Reagan, R., "Fiber Connectors Reaches New Levels," Connector Specifier, March 2003.

Reagan, R., C. Hill, "Continuous Improvement in Optical Connectors," NFOEC Proceedings, September 2002.

Leshin, B., R. Reagan, A. Wood, J. Quinlan, "Optical Surveillance in a Metropolitan Network," NFOEC Proceedings, September 1999.

Reagan, R., B. Leshin, C. Chiburis, K. Goddsell, "Intelligent Fiber Cable Testing Systems," NFOEC Proceedings, July 1997.

Michaels, P, R. Reagan, C. Hill, R. Zelins, "Requirements for High Density Fiber Frames," NFOEC Proceedings, July 1996.

Reagan, R., R. Jensen, R. Travieso, "Connectorizing the Broadband Network," NFOEC Proceedings, July 1995.

Reagan, R., C. Hill, B. Leshin Administration for Optical Facility Management," NFOEC Proceedings, July 1995.

Patents:

US 7,489,849, "Fiber Drop Terminal," Reagan, R., J. Gniadek, M. Noonan, M. Baren-Boyem, February 10, 2009.

US 7,477,824, "Universal Bracket for Mounting Drop Terminals," Reagan, R., J. Cobb, January 13, 2009.

US 7349,605, "Fiber Breakout with RFI Device," Noonan, M., R. Reagan, Y. Lu, J. Gniadek, March 25, 2008.

US 7,471,869, "Equipment Layout for Fiber Hubs," Reagan, R., J. Gniadek, T. Parsons, M. Noonan, December 30, 2008.

US 7,460,759, "Post Mount Fiber Distribution Hub," Reagan, R., J. Cobb, T. Leblanc, J. Gniadek, December 2, 2008.

US 7,400,816, "Telecommunication Apparatus for Distribution," Reagan, R., J. Gniadek, T. Parsons, M. Noonan, July, 15 2008.

US 7,369,741, “Storage Adapter with Dust Cap Posts,” Reagan, R., J. Gniadek, T. Parsons, M. Noonan, May 6, 2008.

US 7,245,809, “Splitter modules for fiber distribution hubs,” Gniadek, J., T. Leblanc, Y. Lu, K. Millea, R. Reagan, July 17, 2007.

US 7,200,317, “Systems and Methods for Splitter Pigtail Storage,” Reagan, R. , J. Gniadek, T. Parsons, M. Noonan, April 3, 2007.

US 7,171,102, “Optical Communication Signal Distribution Enclosure,” Gniadek, J., T. Leblanc, Y. Lu; Yu, K. Millea, R. Reagan, January 30, 2007.

US 7,146,089, “Fiber Distribution Hub Administration Method,” Reagan, R., J. Gniadek, T. Parsons, M. Noonan, December 5, 2006.

US 7,103,255, “Optical Splitter Module,” Reagan, R. , J. Gniadek, T. Parsons, M. Noonan, September 5, 2006.

US 7,088,899, “Configuring Pigtails in a Fiber Hub,” Reagan, R., J. Gniadek, T. Parsons, M. Noonan, August 8, 2006.

US 6,983,095, “Fiber Distribution Hub System and Method,” Reagan, R. , J. Gniadek, T. Parsons, January 3, 2006.

US 6,839,498, “Optical Cable Swivel for Fiber Distribution Frame,” Janus, N, W. Parzygnat, A. Pellegrino, R. Reagan, January 4, 2005.

US 6,445,865, “ Optical Fiber Jumper Cable Bend Limiter,” Janus, N, A. Pellegrino, R. Reagan, September 3, 2002.

US 6,385,381, “Fiber Optic Interconnection Combination Enclosure,” Janus, N, A. Pellegrino, R. Reagan, May 7, 2002.

US 6,321,017, “Portal bend limiter/strain reliever for fiber closure,” Janus, N, A. Pellegrino, R. Reagan, E. Green, November 20, 2001.

US 6,256,445, “Illuminated Optical Connection Port,” Jennings, M., F. Leone, V. Schumakoff, R. Pimpinella, R. Reagan, July 3, 2001.

US 5,956,832, “Optical Connector Insertion Tool,” Reagan, R., September 28, 1999.

US 5,838,845, “Primary Stage Optical Switch Assembly,” Leone, F., R. Pimpinella, R. Reagan, November 17, 1998.

US 5,793,909, “Optical Monitoring and Test Access Module,” Leone, F, J. Mock, J. Palmquist, W. Parzygnat, R. Pimpinella, R. Reagan, August 11, 1998.

US 5,784,516, “Optical Fiber Distribution Shelf Assembly,” Parzygnat, W, R. Pimpinella, R. Reagan, July 21, 1998.

US 5,724,468, “Fiber Distribution Shelf Electronic Back plane,” Leone, F., W. Parzygnat, R. Pimpinella, R. Reagan, March 3, 1998.

US 5,694,511, “Optical Switching Apparatus and Method Pimpinella, R., R. Reagan, December 2, 1997.

US 5,689,604, “Fiber Optic Operations Center,” Janus, N., F. Leone, W. Parzygnat, R. Pimpinella, R. Reagan, “November 18, 1997.

US 5,636,138, “Jumper Cable Selection and Routing System,” Gilbert, A., A. Petrunia, R. Reagan, June 3, 1997.

US 5,490,229, “Slidably Mounted Optical Distribution Tray,” Ghandeharizadeh, S. R. Reagan, February 6, 1996.

This dissertation is dedicated to my children James and Kara Reagan,
to my parents Harry and Betty Reagan and to Fran Reagan.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Dr. Sanchoy Das, for his guidance and support throughout the entire research at NJIT. I also thank my other committee members at NJIT including: Dr. Carl Wolf, Dr. Athanassios Bladikas, Dr. Paul Ranky and Robert English, for their support on this project.

I offer my sincere thanks to Mike Noonan for his vision, encouragement, and inspiration and for giving me the opportunity to manage NPD in the most dynamic, challenging, and rewarding environment possible at Fiber Optic Network Solutions. I thank Jack Field for his inspiration and encouragement and for giving me the opportunity to manage NPD with vision and discipline at ADC Telecommunications. I also thank Dr. Richard Pimpinella for his encouragement on this project and for the valuable lessons he taught on how to achieve technical excellence in New Product Development at Bell Laboratories. I thank Elliot Baines for his inspiration, encouragement, and for his patience in teaching me a great deal about engineering and product development at Bell Laboratories. I also want to thank posthumously Dr. Louis Scerbo and John Donegan, my early mentors at Bell Laboratories, for the lessons they taught on how to be an engineer and how to be a new product development manager.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Background

Effective New Product Development (NPD) has emerged as a major strategic initiative and will continue to be the central focus of corporate management in the decades ahead (Cooper 2005). This is especially true in organizations involved with technology intensive NPD that requires highly skilled technical resources and where time to market pressure is a constant reality. Managers responsible for NPD decisions face challenges such as changing market conditions, evolving product definition and increasing competition. At the same time they face constant pressure from within the firm to improve productivity, reduce cost and achieve financial results.

Global spending on R&D is expected to exceed one trillion dollars (R&D Magazine 2008) in 2009 with approximately one half of the R&D spending on NPD. According to the American Productivity and Quality Control benchmarking study (APQC 2003), of the new products launched in the US from 2000 to 2003, just 51 percent were launched on time and only 56 percent achieved desired financial goals. Studies (Cooper 2005) indicate that on average one third of new product developments are considered failures. Observations over the last two decades (More, 1986, Griffin 1996, Cooper 2001) consistently document that technology-based innovations tend to fail in the market because they don't meet customer needs, they are unreliable, too expensive, provide too little benefit or because they are introduced too late and miss the market window.

Comprehensive analysis of best practices (Cooper 2005) indicates that NPD success rates are improved where NPD planning and product innovation strategy is included as an integral part of the overall strategic planning process. Successful organizations leverage competencies and target attractive markets having sufficient scale and growth to achieve financial success. Focus on the new product selection process, on effective portfolio management and developing capabilities and resources is necessary to achieve growth and return on investment. Success rates on individual projects are increased through improved project planning and execution including striving for unique products and focus on pre-planning activity that validates the scope, opportunity, customer needs, business case analysis and technical feasibility of the project.

Organizations achieving higher NPD success more often have instituted process, structure, tools, metrics and a project management discipline with quality measures and controls to ensure adherence. Success is also correlated with focus on the people aspect of NPD including team structure, having dedicated teams and using cross-functional teams. NPD project management and resource assignment are key success factors and are the focus of this dissertation.

The Product Development Process (PDP) is the general description for the process the firm uses to transform product concepts and opportunities into a stream of products that meet the needs of the customers and the strategic goals of the company (Kennedy 2003). The PDP may be viewed and subdivided in three major steps including; 1) Front End Process (FEP), 2) New Production Development (NPD) process, and 3) commercialization process shown in Figure 1.1 (Koen 2005).

Evaluation and initial screening of the concept is typically performed in the FEP process including assessment regarding technical feasibility and a detailed financial prospectus quantifying value to the corporation. The FEP analysis also considers factors such as fit with the current product portfolio and match to the firm's strategic capabilities and resources. Following the FEP a decision is made on whether to continue with the development.

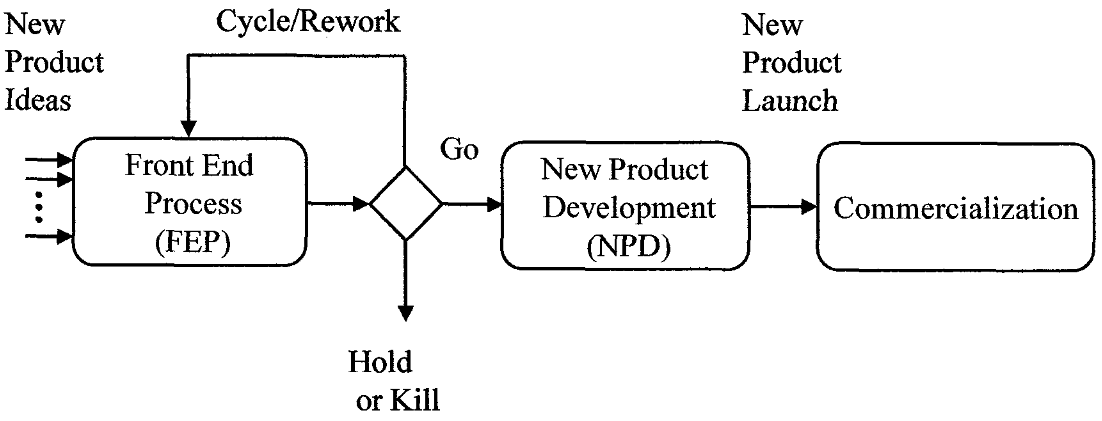


Figure 1.1 Product Development Process (PDP).

Source: Koen, P. (2005), "The Fuzzy Front End for Incremental, Platform and Breakthrough Products." PDMA Handbook, 2nd ed

Following the FEP many ideas may be put on hold pending availability of resources. Other ideas may be screened or "Killed" prior to moving into a formal NPD development phase. PDMA studies (Griffin 1997) concluded a seven-to-one ratio of concepts to new product launches. Data from the 2003 Comparative Performance Assessment Study (Adams-Bigelow 2005) confirmed that it takes seven product ideas (on average) entering the FEP process to achieve one NPD launch.

Using current best practices, the NPD process is often structured as set of sequence of functional steps or stages where each stage has a unique focus. The decision points between each stage act as gates that either allow the project pass to the next stage or interject an alternate or revised plan for the project. The Stage Gate process (Cooper 2001) shown in Figure 1.3 allows an organization to guide a project from one stage to the next using simple decision criteria. At each gate the NPD project is reviewed against the criteria and a decision is made on whether to proceed to the next phase of development.

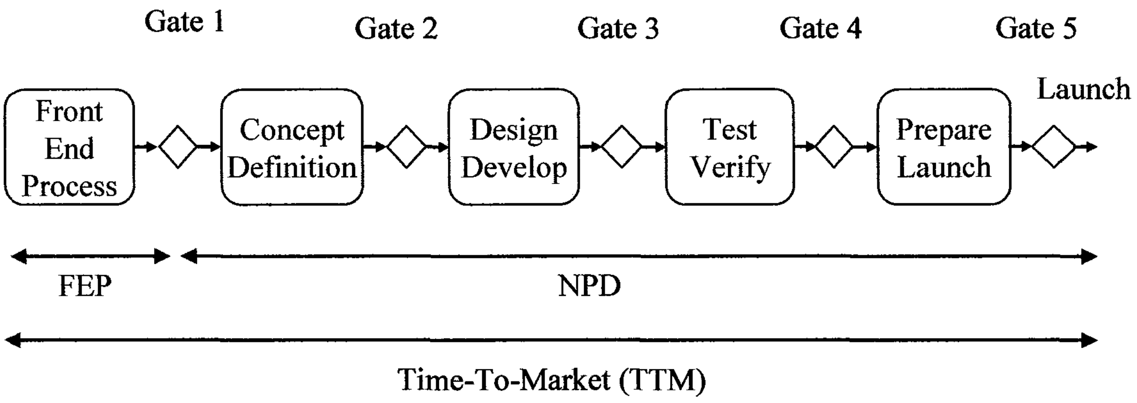


Figure 1.2 Structured New Product Development Process.
Source: Cooper, R.G. (2001). "Winning at New Products," Basic Books.

Time-To-Market (TTM) is measured as the time duration from the inception of the idea to the product launch. NPD processes have evolved over the past several decades and techniques for rapid product development have become increasingly popular throughout the industry (Smith 1998). Significant research to date is aimed at increasing the efficiency and speed of product development. Improvement efforts have often focused on standardized product development processes such as the Stage Gate Process or on design automation tools. In addition project management tools such as project planning via Microsoft Project and application of the Theory of Constraints (TOC) through Critical Chain Project Management (CCPM) have achieved interval reduction

New Product Development continues to change and evolve and more structure and tools have been introduced over time. Initiatives for further improvements have focused on continuous improvement principles (Deming 1986), Total Quality Management (TQM) (Juran 1999) and more recently Six-Sigma methodologies (Pyzdek 2003). The results have been significant improvement in the quality of the manufactured product. However continuous improvement methods have not necessarily translated into improvement in the effectiveness of the product development process itself as evidenced by continued high rate of new product failures where products are late to market and are not meeting customer expectations.

Recent research (Kennedy 2003) indicates that application of the structured product development processes based on techniques derived from TQM and Six Sigma methodologies may not yield further improvements in NPD performance. New product development tends to be iterative by nature therefore imposing more structure may lead to inefficiencies. Alternative approaches such as Lean Product Development (LPD) or Knowledge Based Development (KBD) are gaining interest as possible approaches for continued improvement.

An ongoing challenge for the firm is to balance the right mix of the various NPD projects to achieve the strategic goals. Success rates depend on the product type or newness of the product. Most firms may simultaneously tackle a mixed portfolio of NPD projects resulting in products that are new to the company or new to a market (Crawford 2003). Achieving the mix of projects typically results in resource assignment and scheduling of small, medium and large projects with multiple projects often scheduled simultaneously or in rapid succession.

1.2 Research Scope

NPD research may be broadly categorized in four key functional areas (Cooper 2005) including; 1) Product Innovation and Technology Strategy, 2) Product Portfolio Management (PPM), 3) Project Management and Techniques and 4) Product Development Process Structure. All of the four areas are major drivers of New Product Development performance and all are interrelated.

This dissertation assumes that the NPD organization is operating in a firm where the product innovation and technology strategy has been well established and where the product/project portfolio has been pre-selected based on strategic criteria. This dissertation is specifically targeted at the more tactical problem associated with NPD Project Management once projects have been selected and chartered. The research focuses on project and resource prioritization, resource assignment and scheduling. The research scope considers the complexity associated with assigning technical professionals that vary in discipline type and skill level. The research supports resource assignment decisions in a multiple project environment where multiple products are developed simultaneously or in rapid succession.

The dissertation deals with decision making in a dynamic environment where each project state is continually changing and where decisions must be made quickly to meet time-to-market constraints. Multi-project decisions take into consideration time constraints and resource constraints that directly translate into market success for the new product and financial success for the project.

The research is particularly applicable in development of technology-intensive products such as telecommunications products, information technology products, consumer appliance products, consumer electronics products and computer products. The research is specifically focused on the decision problem associated with resource assignment across Short Life Technology Intensive (SLTI) NPD projects having the following characteristics:

- Projects containing high technology content including systems, electronics, mechanical components, mechanical assemblies, firmware, software, packaging, etc. and dependent on availability of a particular technology or maturity of a technology in order to carry through the development to completion.
- Projects having relatively short development cycles in the range of six (6) to eighteen (18) months.
- Products having relatively shorter life cycle, typically in the range of one (1) to three (3) years, therefore it is critical to launch the product at an opportune time to maximize the commercial success.
- Products launched into stationary market windows where the market opening and expiration dates are well defined and understood and where unrecoverable loss occurs with any delay in the launch date.
- Products are created in an environment where there are usually multiple technical products under development simultaneously leading to resource contention and complexity of scheduling resources of various types and skill levels.
- Projects are carried out in a dynamic environment where priorities are often changing and the risks (of failure) are proportionately higher than for other products having less technical complexity due to the uncertainty of customer requirements and risks regarding availability of technical professionals having the skill type and level required to carry out the tasks.

While particularly focused on SLTI-NPD projects, the research applicability may be extrapolated into any class of products where buyer preferences are refined and where products are introduced in wider variety, with greater frequency and under increasing competition. For instance the research may eventually be helpful in new product development of automotive products, medical products and increasingly other products that in the past have seen less market churn but now are experiencing increased market and competitive pressures.

This research is not focused on the marketing and business management issues associated with sizing the markets, completing the Front End Process and project/product portfolio selection. For the purpose of this research we assume that the Front End Process has been completed and projects have already been selected. All projects have detailed requirements and expectation for achieving revenue and profitability over the product life cycle. Finally the FEP has identified optimal market timing including the optimal product launch date to fully achieve the maximum financial objectives of the NPD project.

1.3 Problem Statement

New product development organizations operating in a multiple project environment face significant challenges in assigning technical staff across multiple projects to ensure the timely completion of projects and product launch corresponding to market windows. The importance of Time-To-Market (TTM) (Griffin 1997, 2002) and launch timing relative to customers and competitors is often cited (Di Benedetto 1999) as a key success factor in NPD. The time-to-market delay consequence for technology intensive products is often characterized by stationary market windows having fixed opening and closing dates. In a stationary market window, the potential value that a product may achieve is determined by the ability to launch the product coincident with the window opening date. Products launched beyond the window opening date result in unrecoverable value lost over the product life-cycle.

The time-to-market value impact is illustrated for integrated circuits in Figure 1.3 (IBM 2006). The chart shows three different types of markets: fast, medium, and slow moving. In a fast moving market, being just three months late can cost over a quarter of a product's potential revenue while being twelve months late can result in loss of over 90% of the revenue. A distinguishing characteristic of the time-value relationship is also apparent in the slower moving market where the cost of delay is less (than the faster moving market) but still significant relative to the maximum potential.

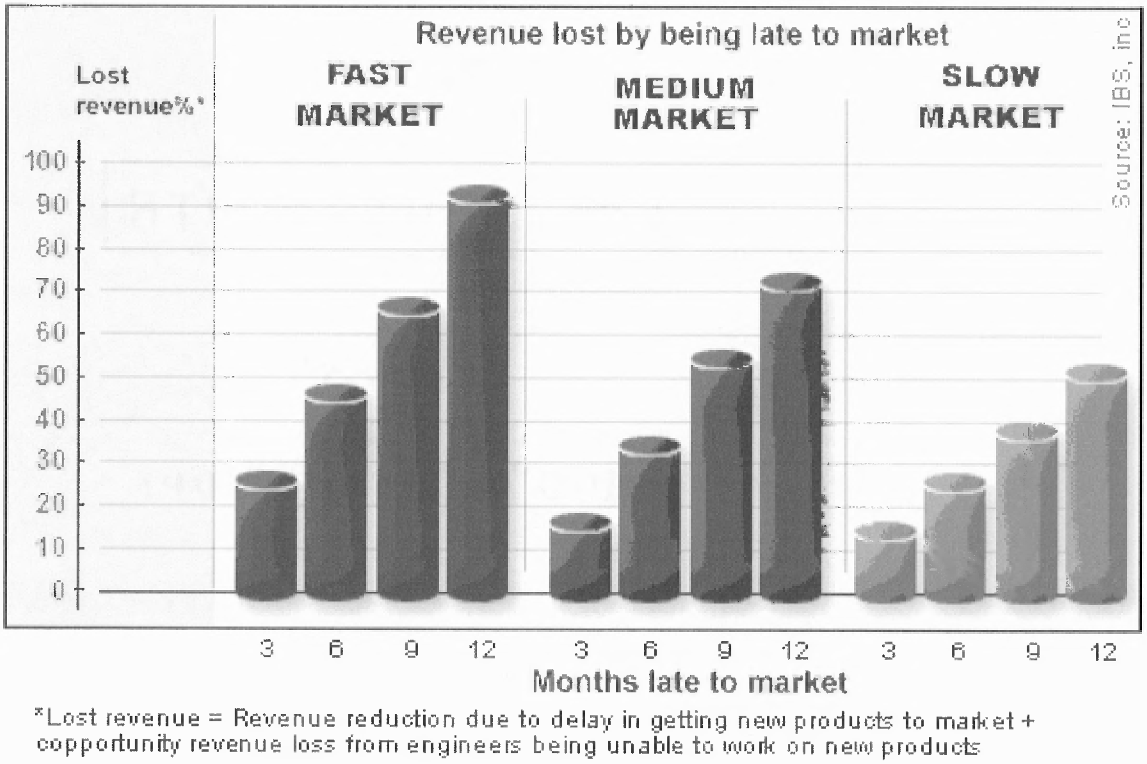


Figure 1.3 Integrated circuit lost revenue due to TTM Delay.

Source, IBM, (2006), ibm.com.

Projects are ideally planned to take advantage of the known market window and to maximize profits by introducing the product at the earliest date the market opens. The cost of delay is the projected value lost due to the product being introduced late relative to the value achieved if the product were introduced at the earliest possible launch date.

The business case prepared at the outset establishes the financial goals and objectives for the product over the potential life cycle. The potential project value may be measured as revenues over the potential life of the product. The potential project value is depicted in Figure 1.4 as the area under the potential product life-cycle curve which increases as the market increases and decreaes as the market decreases. The NPD product and project value assumes a stationary market window that has opening and closing dates that are known and fixed in time. The opening date is the earliest date a product can be sold into the market and the closing date is the last date a product can be sold.

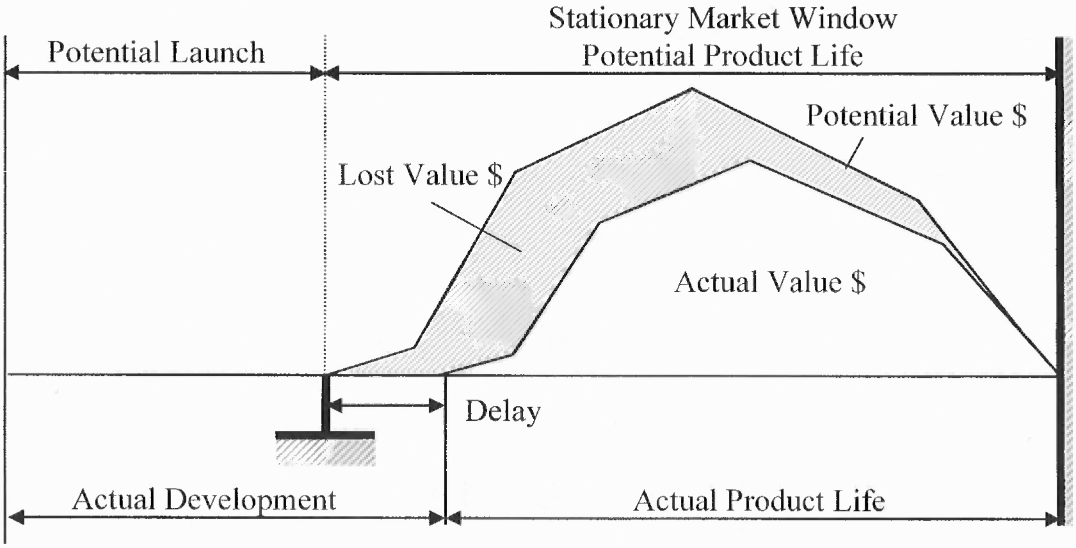


Figure 1.4 NPD project value in a stationary market window.

In such an environment the NPD project management objective is to launch the new products such that the net present value of the product revenues, after factoring in any launch delays is maximized. The corresponding NPD project management decision then is to assign available technical resources to minimize the sum of the revenue loss due to product launch delays across the multi-project portfolio.

Projects often experience delays and actual schedules are longer than required to satisfy potential launch dates. Because the closing date is fixed in time a delay essentially compresses the actual life of the product after it is launched. The unrecoverable lost value is computed as the difference between the areas of the two curves in Figure 1.4. This lost value is the penalty of the project delay typically quantified as lost revenues (or lost profits).

A typical project team may have multiple resource types such as mechanical engineers, electrical engineers, manufacturing engineers and technicians. Resources of differing discipline are generally not interchangeable regarding project role and task assignments. Managing technical NPD projects requires the ability to assign technical professionals of different disciplines and skills to projects and tasks and achieve project objectives within the specified time frame. Furthermore a range of skill levels typically exists for each resource type in the organization. Along with the range of skill levels the proficiency of completing tasks may differ significantly between a resource with a higher skill level and that of one with a lower skill level. The resource assignment decisions must account for differences in skill level and associated work rates.

Most organizations are typically not only concerned with a single project but have multiple projects that must be planned in parallel. Typically organizations also have a queue of NPD projects awaiting resources and undertake projects in rapid succession. The focus of this research is to provide a framework for resource assignment across multiple product development projects. The dissertation develops a model for characterizing the impact of resource assignment decisions on individual projects and the overall program of NPD projects.

At any given point in time, the NPD organization must monitor and review the status of each project, to make any necessary adjustments and make sure the entire program is on track to meet objectives. This will typically happen weekly, monthly or at convenient transition points on the project such as at gate reviews or integrating events. NPD project management may often have very few degrees of freedom to increase the budget or alter the staff size of the organization. In fact resources are usually constrained based on annual budgets and very inflexible.

The problem confronting NPD management is to determine the impact of decisions regarding assigning existing available technical resources on the overall program of multiple projects. There are currently limited means for considering the impact of these decisions in a rapidly evolving SLTI-NPD multi-project environment. In order to consider the impact of the decisions managers must account for all of the highly differentiated technical tasks on each project. In addition, management must consider the availability of technical staff with the resource types and skill levels required to complete the project. Furthermore they must quickly assess the impact of resource assignment decisions on delay and on the projected program value.

Current decision tools typically do not consider the technical complexity of the requirements and resources involved or the inherent variation associated with task duration and schedule duration estimates. There is a need for an improved NPD resource assignment model that improves planning and effectiveness in multi-project, resources constrained, and technologically intensive product development environments.

1.4 Decision Model Structure

This dissertation focuses on the NPD multi-project resource assignment problem for technology intensive products launched into stationary markets. Key components of the resource assignment problem structure are depicted in Figure 1.5. The structure includes a program of multiple projects each consisting of technical requirements that are translated into project tasks. The tasks are characterized by skill requirements, durations and logical precedence relationships. A set of resources are available with appropriate discipline (resource type) to complete the project tasks. The resources have varying skill level with the potential for different work rates and task completion times. Each project has an expected value quantified as the revenues, profits or net present value of the product over the product life cycle. The objective is to complete assignments and schedules such that the value lost due to delay is minimized across the program of multiple projects.

Schedules and project launch dates are estimated as resources are assigned to projects. The value lost to the project is the difference between the potential value and the actual value realized. The key SLTI-NPD project management objective is to launch the new products such that the net present value of the product revenues, after factoring in any launch delays is maximized. The corresponding NPD project management decision then is to assign available technical resources so as to minimize the sum of the revenue loss due to product launch delays across the program of multiple projects.

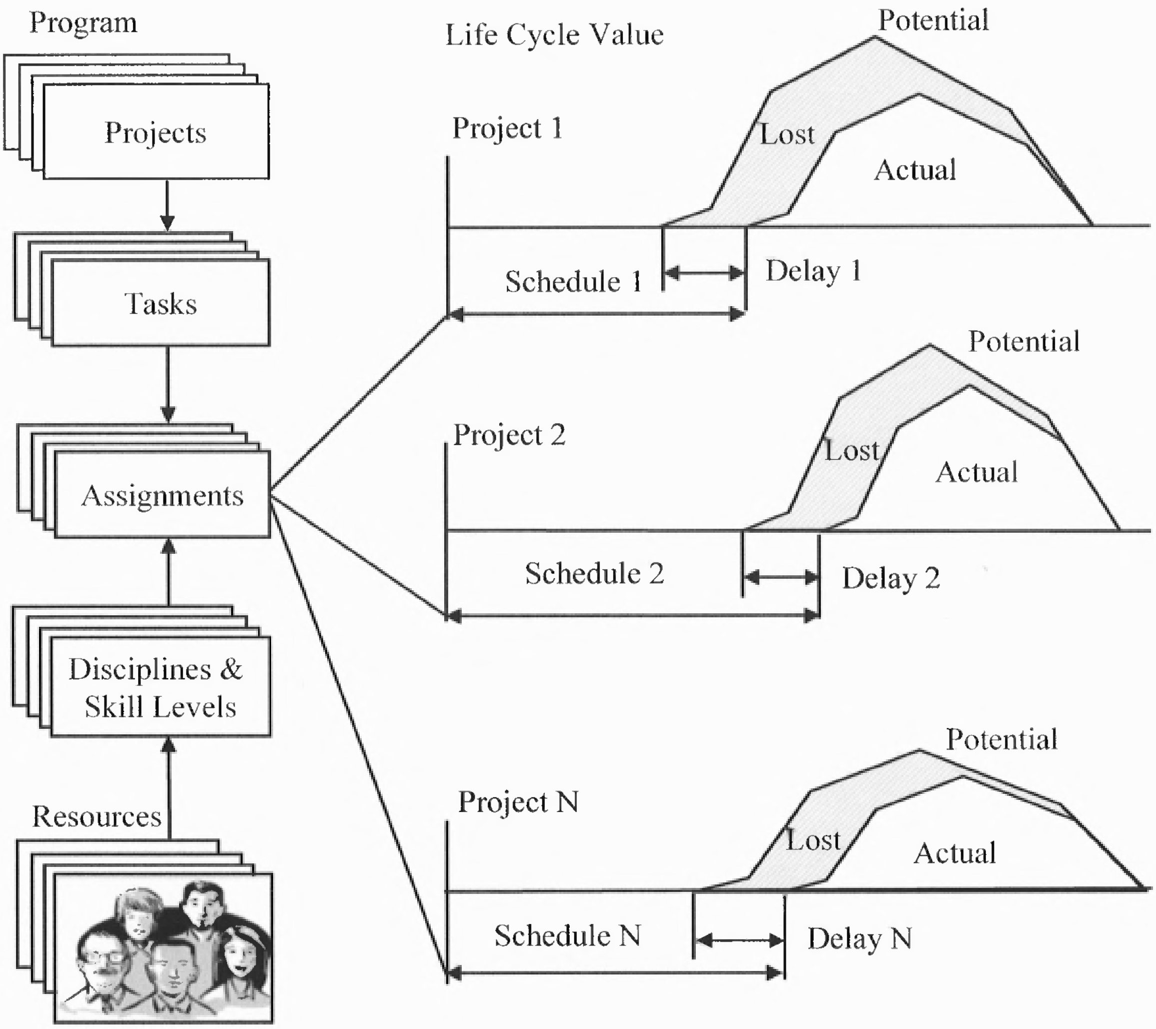


Figure 1.5 NPD multi- project management problem structure.

In order to model the problem for this dissertation we assume that certain key data is known and available, including:

- Potential value for each product.
- Potential life for each product once launched.
- Potential launch date or earliest date a product can be sold.
- Number and identification of each task.
- Resource type required for each task.
- Duration estimates for each task.
- Network/logical precedence relationship for each project.
- An inventory of technical resources of known type and skill level.

1.5 Research Objectives

This dissertation focuses on the SLTI-NPD resource assignment problem with the objective to launch new products to maximize the net present value of the product revenues after factoring in any launch delays. The corresponding NPD project management decision is to assign available technical resources so as to minimize the sum of the revenue loss due to product launch delays across the program of multiple projects. Four objectives are considered:

1. Research Objective Number One (see Chapter 3).

   Characterize SLTI-NPD projects and products using case studies to highlight technical complexity, trend towards shorter development cycles, time-sensitive revenue life cycle and time-to-market success factors.

2. Research Objective Number Two (see Chapter 4).

   Formulate the SLTI-NPD project management resource assignment problem including characterization of projects by requirements, tasks, skill requirements, durations, logical relationships and a known potential revenue cycle. The resources are technical professionals characterized by skill levels in a known set of technical disciplines (resource types).

3. Research Objective Number Three (see Chapter 5 and Chapter 6).

   Develop a decision model and solution for multi-project resource assignment and scheduling using a multi-step process which first prioritizes projects for resource assignment and then assigns and schedules resources to specific tasks across multiple projects while minimizing the loss due to schedule delay.

4. Research Objective Number Four (see Chapter 7).

   Develop a decision support tool for resource assignment in a multi-project environment. The Resource Assignment and Multi-Project Scheduling (RAMPS) decision support tool incorporates the decision model and solves the resource assignment problem two phases; (i) a constructive approach that employs priority rule heuristics to derive feasible assignments and schedules and (ii) an improvement heuristic that considers the productivity gains that may be achieved by interchanging resources of differing skill levels and corresponding work rates.

1.6 Research Significance

The success of corporations and organizations usually depends on the success of new products in the marketplace. On average one third (Cooper 2005) of new products are unsuccessful and launch delay is often cited as significant factor contributing to failure. New Product Development (NPD) organizations face significant challenges assigning and scheduling resources of various technical disciplines and skill levels to multiple projects while achieving market driven launch dates that minimize revenue cycle loss across the entire program of projects. This dissertation addresses specific aspects of the resource assignment problem crucial to NPD success, including:

1. A scheduling methodology formulated with critical chain project management techniques that incorporates two-point task estimating to simplify consideration of task and project schedule variation while still providing robust baseline scheduling estimates. Traditional current best practices typically use single point estimates (deterministic) resulting in unrealistic schedules without consideration of estimating variance. Triple-point estimates (PERT) while incorporating variation are infrequently used in practical NPD environments due to the extra effort in developing the full range of data required.

2. A product life-cycle valuation function anchored to a pre-defined launch date provides linkage between robust schedules and time-sensitive revenue life cycle and models the loss of revenue at an accelerated rate which is consistent with the rate of revenue loss commonly seen in technology intensive products. Traditional models tend to model the time-sensitive loss as a linear function resulting in a high risk of underestimating the impact of launch.

3. A project prioritization scheme that uses the product life-cycle valuation function and robust schedule as the basis for deriving productivity metrics. The productivity metric provides management with parameters to adjust based emphasis based on knowledge (optimism/pessimism) of overall resource availability to the program. The productivity metric is used to rank projects for resource scheduling. Traditional best practices base priority decisions strictly on NPV without consideration to the effort needed to complete the project. While NPV remains an excellent metric the research shows that productivity metrics provide improved guidance for resource assignment.

4. A resource prioritization scheme based on pre-sorting resources by skill level prior to assignment. Prioritized skill level assignment results in substantial improvement in the scheduling performance and improved efficiency in scheduling algorithms.

5. A resource assignment and scheduling solution that includes; 1) a constructive approach that employs priority rules for projects, tasks and resources to derive a feasible solution and 2) an improvement heuristic that considers the improved productivity that can be achieved by interchanging resources of differing skill levels and corresponding work rates.

6. A Resource Assignment and Multi-Project Scheduling (RAMPS) decision support tool that executes the entire methodology and provides automated support for resource assignment and scheduling in a multi-project environment. The RAMPS tool first prioritizes projects for resources assignment and then assigns and schedules resources to specific tasks across multiple projects while minimizing the revenue loss across all projects. Current project management tools typically provide little or no automated support for multiple projects, multiple resource types, project prioritization, resource skill level, resource prioritization, assignment and financial modeling.

7. A computational methodology is introduced including multiple non-dominated project data sets that allow simulation of realistic project cases. Results point to significant productivity and efficiency gains that can be achieved through effective planning, prioritization and deployment of technical resources.

The research has the potential to improve NPD resource assignment decisions allowing the NPD organization to more effectively and efficiently deploy resources. One result will be to improve the throughput and cycle time of New Product Development. The research also has the potential to improve success rates which will lead to improved financial performance across the program of multiple projects.

1.7 Dissertation Overview

The dissertation presents the development of the RAMPS methodology for resource assignment in SLTI-NPD projects. The RAMPS steps can be divided into two phases. The first phase focuses on project valuation and prioritization, while the second phase focuses on resource assignment. This dissertation is organized into eight chapters that describe the overall model, specific RAMPS steps, experimental analysis and results.

Chapter 2 reviews the NPD literature in the context of NPD project management, project prioritization and selection and resource constrained project scheduling. Chapter 3 provides a case study review of Short Life Technology Intensive (SLTI) NPD projects and highlights technical complexity, short development cycles, short revenue life-cycles and time-to-market success factors. Chapter 4 develops a formulation for the NPD project management resource assignment problem including task formulation, skill characterization, resource inventory and the resource assignment decision variables. Chapter 5 develops a productivity based project prioritization model with emphasis on minimizing the lost project value as a function of anticipated delay relative to potential launch date. Specifically, the RSM, PVM and PPM modules depicted in Figure 1.6 are presented in Chapter 5. Chapter 6 provides the resource assignment method that employs a combination of traditional project management techniques with iterative search algorithms and rules based heuristics. Specifically, the RPM, CAM and SLAM modules depicted in Figure 1.6 are presented in Chapter 6. Chapter 7 describes the RAMPS Decision Support Tool and also provides a computational analysis using project data sets that simulate actual multi-project programs. Chapter 8 provides discussion, highlights areas for potential future research.

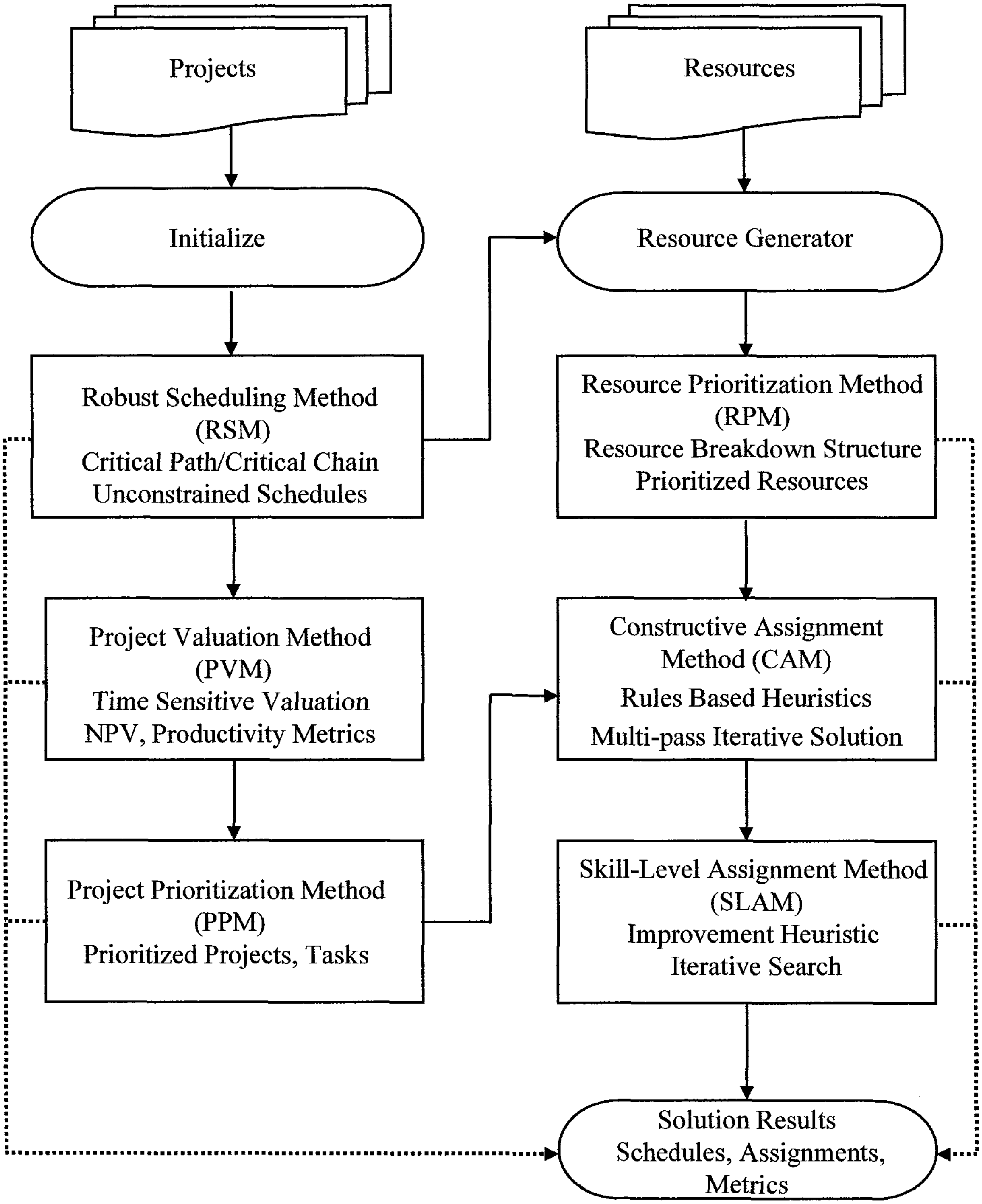

Projects
Resources
Initialize
Resource Generator
Robust Scheduling Method (RSM)
Critical Path/Critical Chain
Unconstrained Schedules
Resource Prioritization Method (RPM)
Resource Breakdown Structure
Prioritized Resources
Project Valuation Method (PVM)
Time Sensitive Valuation
NPV, Productivity Metrics
Constructive Assignment Method (CAM)
Rules Based Heuristics
Multi-pass Iterative Solution
Project Prioritization Method (PPM)
Prioritized Projects, Tasks
Skill-Level Assignment Method (SLAM)
Improvement Heuristic
Iterative Search
Solution Results
Schedules, Assignments,
Metrics

Figure 1.6 RAMPS Methodology for SLTI-NPD Resource Assignment

CHAPTER 2

LITERATURE REVIEW

This chapter provides a review of NPD literature in the following areas 1) NPD overview, 2) technology intensive products, 3) project prioritization, 4) NPD project management and 5) resource constrained project scheduling.

2.1 NPD Literature Overview

Krishnan (2001) provides a comprehensive overview of the NPD literature with a broad coverage of product development research including citations to over two hundred references in the academic fields of marketing, operations management and engineering design. This literature review focuses on product development projects and decisions within a single firm. Krishnan's review provides references in the areas of concept development, supply chain design, product design, performance testing and production ramp-up and launch. A complete set of references is included that discuss NPD speed, price, capacity, reliability, etc. References cited in Krishnan are systematically drawn from the following journals.

- Journal of Product Innovation Management
- Management Science
- Marketing Science
- Journal of Marketing Research
- Research Policy
- Strategic Management Journal
- IEEE Engineering Management
- Research in Engineering and Design
- ASME Journal of Mechanical Design

The field of NPD is rapidly evolving and extensive resources, current research and links are available at the following web sites:

Table 2.1 NPD Web Site References

Web Site	Description	Sponsor
www.pdma.org	Journal of Product Innovation Management, Body of Knowledge, latest research on NPD performance	Product Development Management Association
www.npd-solutions.com	Product Development Forum, NPD Papers, Resources & Links	DRM Associates
www.prod-dev.com	Stage-Gate, Portfolio Management, Working Papers	Product Development Institute
www.roundtable.com	NPD Best Practices, Lean Product Development, Portfolio Management	Management Roundtable
www.ulrich-eppinger.net	Product Design and Development, extensive links	Eppinger & Ulrich
www.newproductdynamics.com	New Product Development and Risk Management Publications, Links	New Product Dynamics
cipd.mit.edu	Center for Innovation in Product Development, MIT Research	MIT

Excellent textbooks are also available in the field of NPD research:

Table 2.2 NPD Text References

Title	Year	Author
New Products Management	1983	Crawford & Di Benedetto
The Machine that Changed the World	1990	Womack, Jones and Roos
Product Design and Development	1995	Eppinger & Ulrich
Design for Excellence	1996	Bralla
Developing Products in Half the Time	1998	Smith & Reinertsen
Thinking Beyond Lean: How Multi-Project Management is Transforming Product Development at Toyota and Other Companies	1998	Cusumano & Nobeoka
Product Strategy for Technology Companies	2001	McGrath
Winning at New Products	2001	Cooper
Portfolio Management for New Products	2001	Cooper, Edgett, & Kleinschmidt
Proactive Risk Management: Controlling Uncertainty in Product Development	2002	Smith & Merritt
PDMA Tool book for New Product Development	2002	Belliveau, Griffin & Somermeyer
Product Development for the Lean Enterprise	2003	Kennedy
Project Management: A Systems Approach to Planning, Scheduling and Controlling	2003	Kerzner
PDMA Tool book 2 for New Product Development	2004	Belliveau, Griffin & Somermeyer
PDMA Handbook of New Product Development	2005	Kahn, Castellion, & Griffen
Product Leadership	2005	Cooper
The Toyota Product Development System	2006	Morgan & Liker

2.2 Technology Intensive Products

Technology intensive NPD (Gertler 1995) is quite different from that in most consumer markets. For technology intensive products seller-customer interactions take on the characteristics of a collaborative relationship. In mature industries like consumer goods, NPD is usually a seller-led process where individual customers play a relatively passive role during most phases of the NPD process. In contrast, the development of technology intensive products is characterized by close and frequent interactions between sellers and buyers that extend through multiple phases of the process.

A balance between market and technology focus is needed (Crawford 2003, Gatignon 1997) for successful technology intensive product innovation to occur. The level of interaction between R&D and market orientation will influence the characteristics of technology innovation including type of innovation (e.g. incremental vs. radical), speed of innovation, capacity for innovation and quality of innovation.

The new competitive landscape (Hitt 1998) describes an environment driven by technological intensive product revolution and significant globalization, that is moving towards hyper competition (rapidly escalating competition and strategic maneuvering), extreme emphasis on price, quality and satisfaction of customer needs, and an increasing focus on innovation (both in technology and new products). Furthermore time frames of all strategic actions are being reduced. In this new competitive landscape firms exist in highly turbulent and chaotic environments.

The importance of Time-To-Market (TTM) (Griffin 1997, 2002) and launch timing relative to customers and competitors is often cited (Di Benedetto 1999) as a key success factor in technology-intensive products NPD. Product life cycles (Bettis 1995) in certain markets have significantly shortened thereby compressing the available time window for recouping the expenses associated with technology based product development. NPD speed is critical in technology intensive products (Alfonso 2008) because product life-cycles are shrinking and obsolescence is occurring quicker than in the past, while competition has also intensified. The time-to-market delay consequence is important for an entire class products characterized by technical complexity. Products such as consumer electronics (MacCormack 1997) that include real time operating systems, hand-held devices, palm devices, cell phones, and mobile communication systems require return on investment within a short time span of about two years.

The life expectancy of computer hardware (Rehemtulla 2006) is changing rapidly based on technology innovation. For example, PC computer technology relatively short life expectancy, since newer technology is being introduced at a rapid pace, and older hardware is dropping out of a vendor's product line. It is common for today's PC vendors to introduce new product or product bundles every two to four months.

2.3 Project Prioritization

The NPD portfolio selection process aligns product development projects with the strategic intent of the organization (Cooper 2001) and provides clear priorities to guide development activity. Planning initiatives result in opportunity statements, business cases, requirements, market driven project plans and high level schedules and targets for a set of NPD projects roughly sized to the capabilities of the organization. This strategic plan provides the potential market value for each new product and identifies the launch date that must be achieved in order to capture the potential life and value for the product.

NPD project selection research generally describes mathematical programming techniques such as Linear Programming or Integer Programming (Loch 2001, Dickinson 2001) for selecting an optimal set of projects. Real Options Theory (Huchzermeir 2001) may be used to make portfolio selections incrementally based on the most recent information available. The Analytical Hierarchical Process (Calantone 1999, Chin 2008) is a simple, intuitive and yet powerful method for project selection through pair wise comparison. Statistical methods (Butler 2001) consider multi-attribute and performance measures to complete ranking and selection of projects. Scoring models (Cooper 2005) account for the many qualitative factors utilized to calibrate a project's desirability, attractiveness, or potential for success.

Financial techniques (Cooper 2005) such as expected commercial value, Net Present Value (NPV), bang-for-the-buck and productivity index are widely used techniques primarily due to ease of use. Additional contributions focus on optimizing the financial returns (Loch 2002) of a NPD portfolio using an analysis such marginal return on investment. A case is made (Ross 1995) that Internal Rate of Return (IRR) and profitability index (PI) provide more efficient capital allocation metrics for project selection versus NPV. However NPV continues to be the dominant factor (Haley 1995) in project selection due to ease of use and universal acceptance. Newer models incorporate the dynamic nature of portfolio management such as changing market conditions and evolving risk management. The dynamic models can be applied to the overall portfolio and may be used to roll up the collection of projects into an overall optimized financial metric.

2.4 NPD Project Management

Increased NPD complexity is generally associated with (Griffin 1997) increased cycle times due to the increased number of functions designed into the product. Significant research (Eppinger 1994) is aimed at structuring complex design projects into sub-tasks in order to develop better products more quickly. A matrix representation captures both the sequence of and the technical relationships among the many design tasks to be performed. These approaches offer opportunities to speed development progress by streamlining the inter-task coordination.

Development of technology intensive products typically requires collaboration across multiple organizations to facilitate knowledge growth (Davenport, 2003), knowledge sharing and dissemination. For example, in developing a consumer electronics product, it is common to employ outside expertise in specialized areas such as Digital Signal Processing, protocols or operating systems.

In planning the NPD project (Kerzner 2003) the project manager must structure the work into small elements that are manageable, independent, integratable and quantifiable. This first major step in the planning process (after requirements definition) is the development of the Work Breakdown Structure (WBS) which is subdivision of the requirements in accordance with the way the work will be performed. The WBS acts as the primary vehicle for breaking the work down into smaller elements, thus providing a greater probability that every major and minor activity will be accounted for. In setting up the work breakdown structure tasks have clearly defined start and end dates and estimates for the amount of effort to complete the tasks.

NPD project scheduling has evolved out traditional project scheduling (Hillier 2001) techniques including the Performance Evaluation and Review Technique (PERT) and the Critical Path Method (CPM). These project scheduling algorithms are derived from networks of sequential tasks each having varying start and end dates and with intricate interdependencies between tasks. Software tools have evolved to facilitate utilization of the scheduling approaches including the popular Microsoft project tool. Other simpler tools such as Gantt charts may be effective for smaller products.

The Theory of Constraints (Goldratt 1997) includes a methodology for resource assignment that involves providing statistical estimates for task completion including an estimate of average completion time and low-risk completion time. These estimates are used in computing the need for probabilistic slack time in the projects thereby providing a practical approach in managing projects via critical chain buffer management.

Analogous research on the resource assignment problem is found in the fields of manufacturing and computing. For instance the cellular manufacturing problem (Askin 2001) involves a methodology for the assignment of workers into cells to optimize the cell-based production environment. This research (Norman 2002) has been extended to include consideration for worker skill type and skill level prior to assigning the workers to the cell. The manufacturing resource assignment problem typically has the objective to maximize throughput or minimize cost. Other similar analogous models are found in the computing industry in order to solve the problem (Lewis 2004) of assigning a set of tasks to one of several processors where the associated inter-task communication and processing costs are minimized.

2.5 Resource Constrained Project Scheduling

Resource constrained project scheduling problem (RCPSP) has been discussed in the literature for several decades (Davis 1975) and is still an active area of research for applied mathematical programming. A vast majority of the research (Van deVonder, 2008) efforts in recent years have concentrated on the development of exact and heuristic procedures for generation of a workable baseline schedules. Exact methods (Herroelen 1998) applied so far include dynamic programming, zero-one integer programming and implicit enumeration with branch and bound. Due to the fact that the RCPSP is one of the most intractable problems in Operations Research it is a popular problem for approximate techniques, including virtually all local search paradigms. Recent contributions focus on Branch and Bound (DeReyck 1998), Genetic Algorithm (Chen 2009) and priority rule heuristics (Xu 2008) to solve the RCPSP. A comprehensive experimental test was recently conducted (Kolish 2006) of more than thirty of the best published algorithms using relatively standard accepted procedures and standard data sets. The results indicate the best algorithms combine several components including forward-backward improvement technique to improve schedules that have been constructed from metaheuristics such as Genetic Algorithms and tabu search. Several excellent texts have been published (Demeulemeester 2002, Klein 2000, Schwindt 2005) that provide broad coverage of mathematical formulations and computational approaches for the single project RCPSP. The

By comparison to the single project RCPSP literature, research on the multi-project scheduling problem is relatively small. The research does date to the 1960s (Fendly 1968) where solutions were developed using priority rule heuristics and Monte

Carlo simulations. Investigation into heuristics (Kurtulus 1982, 1985) discovered that the best rules established for single project scheduling do not necessarily serve as the best priority rules for multiple project scheduling. Furthermore research (Dumond 1988) concluded that no one heuristic will give the best solution under a wide range of simulations conducted. The aspect of differentiated skill levels was introduced (Bock 1990) as component of the multi-project new product development scheduling problem and again focused on priority rules which provided the best results in computational experiments. Extensive computational experiments (Lawrence 1993) tested a wide variety of priority rules related to minimizing project delay.

Recent research continues the path of exploring heuristics (Lova 2000, Kruger 2009) and computational experiments to understand the strength of the various priority rules. In addition search techniques such as Genetic Algorithms (Goncalves 2008) Branch and Bound (Chen 2009) to the multi-project scheduling problem have recently been published demonstrating solutions with excellent results.

CHAPTER 3

SHORT LIFE TECHNOLOGY INTENSIVE (SLTI) PRODUCTS

This chapter satisfies Research Objective Number One (1) by characterizing Short Life Technology Intensive (SLTI) Products using case examples to identify attributes that distinguish SLTI projects from other types of NPD projects. The chapter identifies technical complexity, shorter development cycles and shorter product life cycles as key characteristics of SLTI products. The analysis highlights time-to-market as a key success factor for SLTI products.

3.1 Product Life Cycle

The product life cycle is defined as the time from the introduction of the product until the decline or removal of the product from the market. Products typically progress from one stage to the next during the time period when they are available for sale to the market. The total sales and profits of the product over the life cycle tend to follow a classic "S" shaped curve (Armstrong 2003) as shown in Figure 3.1.

During the product life cycle sales vary from zero in the development stage, to a low rate of change in the introductory stage, a high rate of change during the growth phase, reach a peak during the market phase, and then decrease during the decline phase. In the introduction stage, sales tend to be low because of the new idea or product is just being introduced into the market for the first time. In the growth stage, the sales tend to grow very rapidly and the profits begin to rise. Competitors may see this opportunity and enter the market by copying or improving upon the product. Profits begin to decline because of the increased competition that the new product creates. In the maturity stage the sales may peak or level off. The industry profits continue to decrease during the maturity phase. During the sales decline phase products are typically replaced by newer products. The competition from the declining products becomes more energetic. Firms with strong positions may make profits until the end of the sales decline because they have successfully differentiated their products from the others. Firms may also keep some sales by appealing to their loyal customers or those who may be slow to try the new ideas.

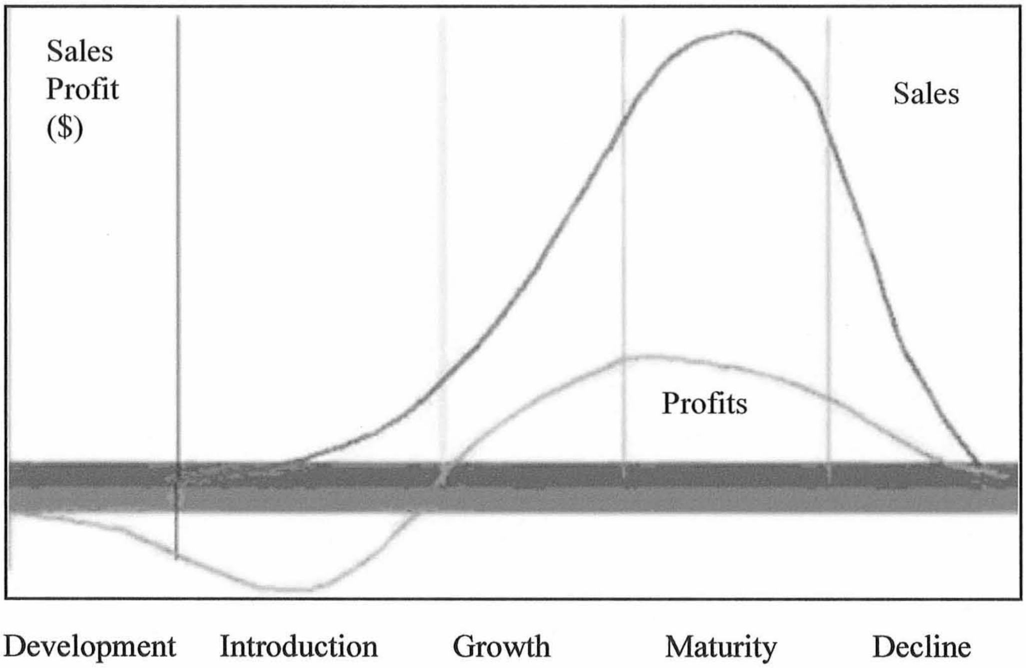


Figure 3.1 Product Life Cycle “S” curve.

Source: Armstrong, G., Kotler, P. (2003), “Principles of Marketing.” Prentice Hall, 10th ed.

An aggressive strategy adopted by some firms is to increase overall sales volume by rapidly introducing new products in rapid succession (Metcalf 2006) and thereby purposely reducing the life cycle of preceding products. One result is to reduce the time between repeat purchases, (referred to as shortening the replacement cycle) which may generate repeat sales in rapid succession. Some firms pursue this strategy to offset the additional costs of research and development and opportunity costs of existing product line cannibalization. In a competitive industry, this can be a risky strategy because consumers may decide to buy from your competitors. However speed of introducing newer and better features may also allow the firm to outpace their competitors.

Products containing high technology are often rushed to market and the initial release may not incorporate all of the desired features. Some features may be dependent on technological advances that are still maturing and not proven adequately for a first generation. However technologies may become available for use in a next generation product. Another common scenario is that resources are not available to complete all the desired features to meet a market window that is opening. In this case the development may be divided into a series of feature releases. This path of rapid innovation and planned obsolescence may be a powerful approach in a fast paced competitive industry.

3.2 Short Life Technology Intensive NPD Projects

The research in this dissertation will focus on Short-Life-Technologically-Intensive (SLTI) NPD products. SLTI products and projects have the following characteristics:

- Products containing high technology content including systems, electronics, mechanical components, mechanical assemblies, firmware, software, packaging, etc.

- Products dependent on availability of a particular technology or maturity of a technology in order to carry through the development to completion.

- Risk (of failure) associated with the technical product being proportionately higher than for other products having less technical complexity due to the uncertainty of customer requirements and rapid product obsolescence.

- Products and projects require availability of technical professionals having a variety of disciplines required to carry out project tasks.

- Products are created in an environment where there are usually multiple technical products under development simultaneously leading to resource contention and complexity of scheduling various technical resources involved.

- Market opening and market expiration dates are stationary, well defined and understood.

- Product life cycle tends to be shorter therefore time-to-market is critical to introduce the product at the earliest date possible in order to maximize the revenue cycle.

- Product features may be introduced in rapid succession through a series of next generation designs either building on the previous generation or replacing the previous generation with a new and improved product.

- Products tend to be pulled by specific customer needs as opposed to those driven by fundamental research.

SLTI products differ from non-technology products in the following ways:

- Technological products require highly skilled professionals such as mechanical engineers, electrical engineers, industrial engineers etc.

- Each of these skilled professionals typically has a specialty or area of concentration that must be categorized and understood so that projects can be planned to carry out work in a reasonably efficient manner.

- Since the supply of highly skilled professionals is inherently limited, these technical resources become the primary constraint in product development.

- Since technology is changing and evolving, market windows are opening and closing quickly and therefore product development organizations are under pressure to reduce the intervals for developing and introducing new products.

- The development of technology intensive products is highly competitive in nature and organizations are increasingly using cycle-time reduction as a means to increase likelihood of success.

- The risk of technology complexity or maturity is not always incorporated into the project management process.

- The risk due to the availability or unavailability of technical professional resources of differentiated skill type and skill level is not always accounted for in the resource assignment process.

- The risk associated with variation in task duration estimates and project schedule estimates is often not considered as a factor impacting potential NPD project value.
- Technology intensive products are typically developed in a dynamic multi-project environment where conditions are constantly changing and where scarce technical resources are typically over assigned thus leading to serious contention and delay.

3.3 Case One - Apple iPod

On October 23, 2001 Apple announced the launch of the iPod offering a whole new category of digital music player to the consumer market. The iPod allows a user to carry an entire music collection in their pocket and listen to it wherever they go. The initial iPod NPD cycle was carried out in 6-to-9 months during the year preceding the NPD launch. Apple employed a team of about 30 engineers including mechanical designers, electronic hardware engineers and software programmers (Kahney 2004). After the first prototypes were constructed, the project received 100 percent of Steve Jobs (CEO of Apple) attention. The design was scrutinized heavily by Jobs to get the sound quality and usability where he wanted it to be.

The accelerated NPD project was enabled by using collaborative development that included contract design and manufacturing firms (Sherman 2002). The unique design relied on a technology platform and reference design for the MP3 decoder and controller created by a third party, Portal Player. The MP3 controller design was selected by Apple because it was an existing stable design that provided the highest quality sound while also allowing customization. Apples design team created a compacted nested design that added to the MP3 controller a flash memory chip, miniature hard drive, interface controller, interface dial and screen and a power management and battery charging controller. Apple specifically avoided combining functions into an ASIC for the first release and instead specified the use of standard off-the-shelf components to reduce the risk and decrease the time-to-market. Apple also employed the services of an outside firm to assist with the operating system and software interface.

The Apple iPod is the prototypical example of a SLTI product. The iPod products were developed and released quickly and in rapid succession (Reynolds 2006). Rather than resting on the success of the initial iPod products, Apple continued to develop and rapidly introduce new generations of products that were considered competitive to Apple's own previous iPod models .that were still selling well. This product development strategy enabled Apple to achieve a market leadership position in just a few years – this in a market where they previously did not compete. The various generations of iPod classic products is depicted in Figure 3.2 and listed below (Kahney 2005, Wikapedia 2009):

- 1st Generation - October 2001: The first generation iPod product was introduced with 5GB memory holding up to 1,000 songs and playing up to 10 hours of music. Apple sold 125,000 units in the fourth quarter of 2001.
- 2nd Generation - July 2002: The second generation iPod product line was expanded to include download capabilities from a Windows interface and additional models with touch sensitive controls were introduced with up to 20GB of memory.
- 3rd Generation - April 2003: The third generation iPod was introduced to include a USB 2 interface and the ability to download music to a Mac via a new web service iTunes and to provide additional models in storage capacity to 30GB of memory.
- Fourth Generation - July 2004: The fourth generation iPod is introduced featuring extended battery life and the ability to shuffles songs.
- Fifth Generation – October 2005 – The fifth generation, also known as the iPod Video, featured a larger screen and smaller Click Wheel and plays video such as TV shows, podcasts, music videos, and movies purchased from online stores.

- Sixth Generation – September 2007 – A new thinner aluminum sliver body with improved battery life, larger backlit display and a completely overhauled user interface incorporating more graphics.

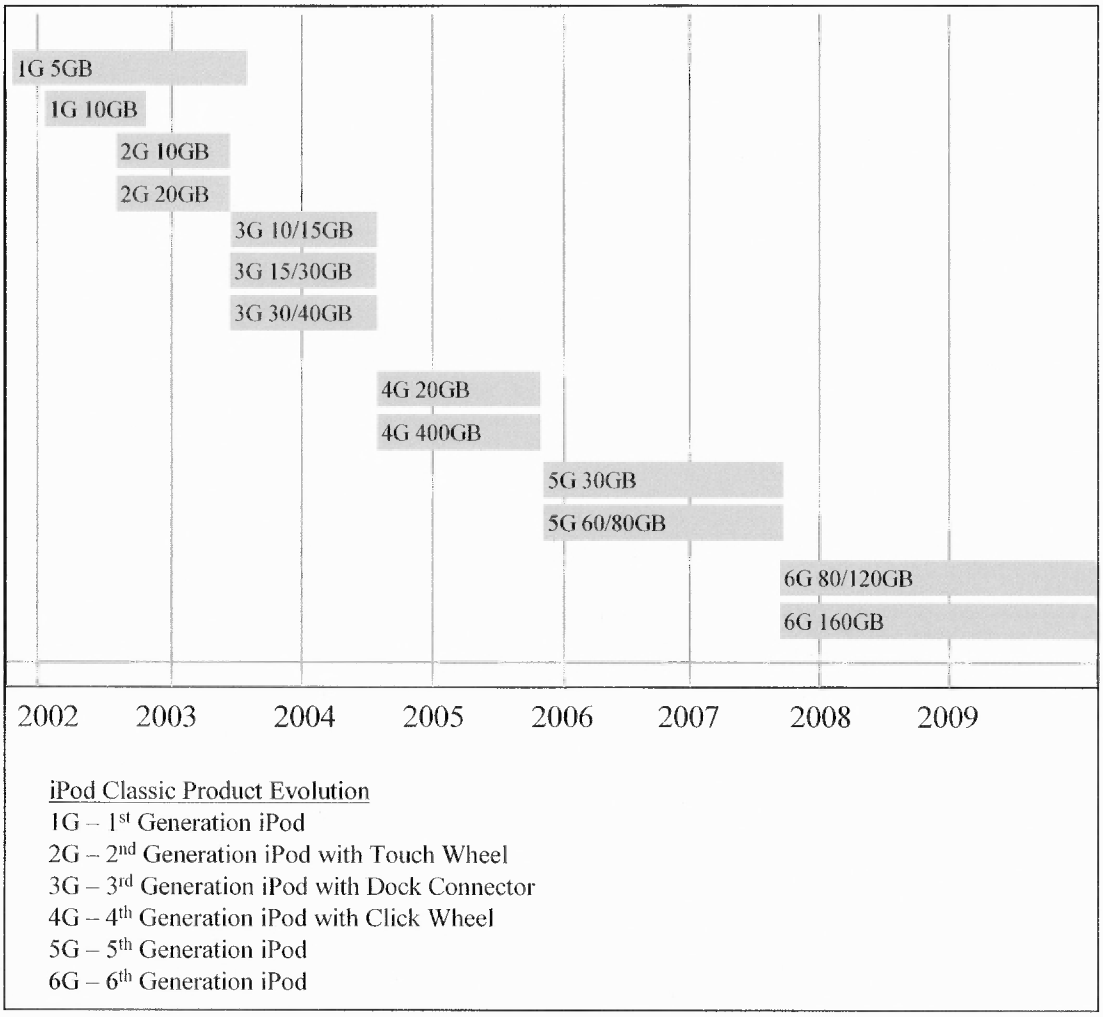


Figure 3.2 Time line for Applies iPod classic.

Source: Wikapedia 2009.

After the original launch in 2001, it took over a year and a half for Apple to hit the one million mark for iPods sold. Six months after launching the third generation iPod, the company had sold its second million iPods. By late October 2004, aided by the release of the fourth-generation iPod, Apple was up to almost 6 million total units. The unit sales trend of the iPod classic is depicted in Figure 3.3.

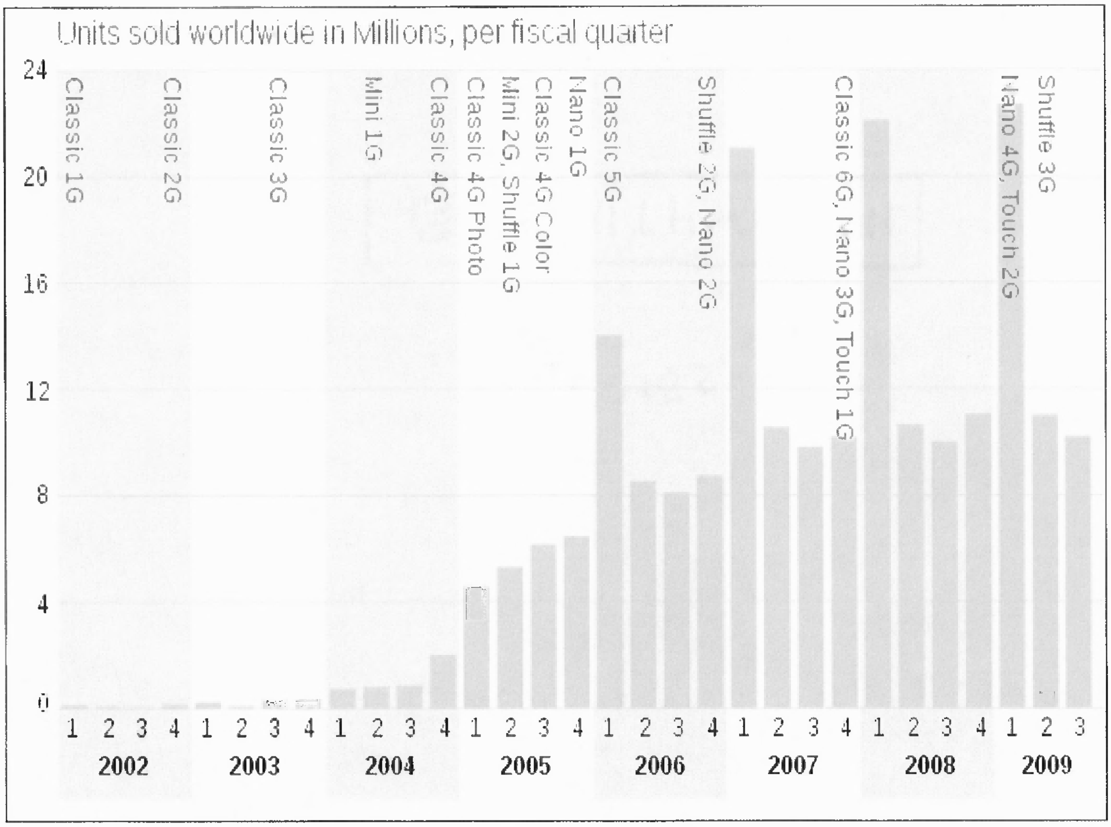


Figure 3.3 Unit sales trend for iPod.

Source: Wikapedia 2009.

Apple iPod developments are excellent examples of consumer electronics SLTI NPD projects environment. As illustrated in Table 3.1, NPD duration averaged 12 months and product life duration averaged only 18 months.

Table 3.1 Apple iPod classic NPD project duration and product life duration.

Generation	**Product**	**Launch**	**Estimated NPD Project Duration** (months)	**Estimated Product Life Duration** (months)
1	First generation	October 2001	9	18
2	Touch wheel	July 2002	9	12
3	Dock Connector	April 2003	12	15
4	Click wheel	July 2004	12	15
5	Video	October 2005	18	24
6	New Profile	September 2007	15	24
Ave.			12	18

Factors contributing to the Apple iPod NPD success are provided in Table 3.2.

Table 3.2 Apple iPod NPD success factors.

Factor	**Apple iPod Development**
Leadership	Involvement of top management in the NPD project leadership.
Team of Skilled Resources	> 30, Mechanical Engineers, Electrical Engineers and Software Engineers dedicated to the project.
Collaborative Development	Contract development of the OS system.
Technology Platform	High performance MP3 decoder available and enabled flexible programming.
Technology Components	All off-the-shelf components reduced risk and time-to-market including; MP3 decoder and control, flash memory chip, miniature hard drive, interface controller/software, interface dial, screen and power management controller.
Differentiation	Ergonomic control interface provided users with a simple way to organize and manage digital music. First to provide practical music download from the internet.
NPD Environment	Multiple project, products replaced in rapid succession.
Time-to-Market	First to market with a product of this type.

3.4 Case Two - FONS Fiber Hub

In July 2002, Verizon issued a Request for Proposal (RFP) to suppliers interested in manufacturing passive optical components for use in constructing a new fiber optic network. The RFP included requirements for a new class of enclosures called the Fiber Distribution Hubs (FDH) that serve as the primary network interface point for connecting fiber optic services to consumers and neighborhoods served by Verizon. The FDH enclosure contains prearranged fiber cables and strands, optical splitters and optical connectors used by Verizon technicians to quickly connect new voice, internet or video services being marketed to residential subscribers.

Fiber Optic Network Solutions (FONS), a small manufacturing firm based in Marlboro, Massachusetts had begun work almost a year earlier on several product platforms aimed at expanding the use of fiber optic components in the outdoor telecommunications environment. One key development included significant technical improvements in the quality, performance and reliability of optical connectors leading to the first industry certification of an optical connector for use in outdoor environments. FONS also developed competencies in packaging optical splitters in re-enterable outdoor enclosures. FONS had a team of six engineers including optical engineers, mechanical engineers and test engineers working on developing the FDH product for Verizon. These advanced technical capabilities allowed FONS to secure a winning proposal and to introduce the FDH product for use in Verizon's new network.

FONS completed the NPD cycle in less than six months and the first of it's kind product was shipped in January 2003 (Reagan 2004). The ergonomic features and design for ease of use made the FONS FDH the favorite of field users. During the first year of deployment FONS FDH had achieved more than 50% market share, shipped nearly 10,000 units. During 2004 FONS quickly began development of a second generation FDH product family to significantly reduce the size over the first generation FDH. The second Generation FDH was developed by the same team and introduced in less than nine months achieving a 50% size reduction along with further ergonomic improvements (Reagan 2005). The second generation FDH products were introduced in phases while achieving more than 70% market share.

The FDH sales continued to increase during the period of 2004 and 2005 establishing FONS as the industry leader. In 2005 FONS once again implemented a rapid NPD cycle that enabled introduction of the third generation FDH product family. The nine month development allowed FONS to introduce the third generation FDH in January 2006 and continue to retain the market leadership position.

FONS FDH Product Developments & Replacements:

- 1st Generation - January 2004: The first generation FDH product was introduced with a partial set of capabilities allowing FONS to capture over 50% market share resulting in shipment of over 10,000 units during the first year of production.
- 2nd Generation - March 2005: The second generation FDH product line was introduced to replace the first generation product line with smaller form factor and a more complete line of options while continuing to include the ergonomic features and benefits found to be successful on the first generation product. The second generation product line enabled FONS to capture nearly 70% market share and increase production to nearly 15,000 units per year in 2004 and 2005.
- 3rd Generation - January 2006: The third generation FDH was introduced to include size reduction and several additional features that make the product even easier to use. FONS continued to demonstrate strong market growth with nearly 80% market share and production increasing to a rate of 20,000 per year in 2006.

The FONS FDH case provides an excellent example of a SLTI NPD project environment. As illustrated in Table 3.3, NPD duration averaged 9 months and product life duration averaged only 15 months. Factors contributing to the FONS NPD success are provided in Table 3.4.

Table 3.3 FONS FDH early NPD project duration and product life duration.

Generation	Product	Launch	NPD Project Duration (months)	Product Life Duration (months)
1	Gen 1	January 2004	7	18
2	Gen 2	March 2005	12	12
3	Gen 3	January 2006	8	15
Ave.			9	15

Table 3.4 FONS FDH NPD success factors.

Factor	FONS FDH
Leadership	Involvement of top management in the NPD project definition and resource assignment.
Team of Skilled Resources	Mechanical Engineers, Optical Engineers and Reliability Engineers assigned to the project.
Collaborative Development	Contract development of the optical splitter technology. Contract manufacture of enclosures and cable assemblies to reach scale.
Technology Platform	Certified Optical connectors available for rugged outdoor environmental applications.
Technology Components	Many off-the-shelf components reduced risk and time-to-market including; optical fiber, optical connectors, optical fanouts and optical splitters.
Differentiation	Ergonomic fiber management design provided compact yet craft-friendly organization.
NPD Environment	Multi-project, products replaced in rapid succession.
Time-to-Market	First to market with a product of this type.

3.5 Case Summary Analyses

The Apple iPod and FONS FDH are entirely different technology-intensive products developed for entirely different markets and applications. However the similarities between the iPod and FDH product development initiatives described the characteristics and attributes of SLTI products and projects. A summary analysis of SLTI NPD project characteristics for the iPod and FDH products is provided in Table 3.5. Key attributes of the iPod and FDH SLTI products and projects include;

1. Technical complexity requiring resources of multiple technical disciplines and skill levels.
2. Short development cycles typically in the range of six (6) to eighteen (18) months.
3. Short product life in the range of one (1) to two (2) years.
4. Next generation products developed in rapid succession thereby obsoleting previous generations with new models.
5. Products developed in competitive markets where TTM is critical to capturing sales and market share that translate into return on investment.
6. Products developed in multiple project environments.

The resource assignment methodology developed in this dissertation is primarily focused on the attributes listed above. It is worth noting that technology strategy and use of established platforms and components played an important role in the success of both the iPod and FDH. In addition the involvement of senior leadership directly in the development details was also found to be a key success factor in both products. Ergonomic design played a major role in allowing both the iPod and FDH to achieve market leadership.

Table 3.5 Analysis of SLTI NPD project characteristics.

	SLTI Product	
Characteristic	**iPod**	**FDH**
Generations Reviewed	6	3
Ave. NPD Duration	12 months	9 months
Av. Product Life	18 months	15 months
Leadership	Heavy top management involvement in the NPD Project.	Heavy top management involvement in the NPD Project.
Resources	Staff with highly differentiated technical skills	Staff with highly differentiated technical kills and levels
Collaborative Development	Use of collaborative development to speed and scale that allowed Apple to outpace competition	Use of collaborative development to provide scale that allowed FONS to outpace competition
Technology Platform	MP3 Decoder	Environmentally rugged optical connectors
Components	Largely off-the-shelf to reduce risk and time-to-market	Largely off-the-shelf to reduce risk and time-to-market
Differentiation	Ergonomic design provided significant advantage over competition	Ergonomic design provided significant advantage over competition
NPD Environment	Fast paced NPD projects, multi-project environment, products replaced in rapid succession	Fast paced NPD projects, multi-project environment, products replaced in rapid succession
Market Type	Consumer Electronics, very competitive (Consumer)	Single customer, business-to-business, very competitive (Industrial)
Time-to-Market	First to market with a product of this type	First to market with a product of this type

The Apple iPod and FONS FDH case studies highlight the importance of launch timing relative to customers and competitors as a key success factor in NPD. These technology intensive products are characterized by stationary market windows having fixed opening and closing dates relative to the market and competitors. In a stationary market window, the potential value that a product may achieve is determined by the ability to launch the product coincident with the window opening date. Following the Apple iPod launch numerous competitors entered the market attempting to derail the iPod market leadership position. Similarly competitors launched products following the FONS FDH introduction. Products launched by competitors beyond the window opening date result in much lower market share and significantly reduced product life-cycle value.

CHAPTER 4

SLTI-NPD RESOURCE ASSIGNMENT PROBLEM

This chapter satisfies Research Objective Number Two (2) by formulating the SLTI-NPD resource assignment problem. The formulation involves an approach to identify and quantify projects, tasks, task durations, logical precedence relationships, project value potential, resources, resource types, skill levels, work rates and assignment decision variables.

4.1 The NPD Project Management Problem

As discussed in Chapter 2, developing Short-Life-Technology-Intensive (SLTI) products requires highly skilled professionals such as mechanical engineers, electrical engineers and industrial engineers. NPD organizations face significant challenges assigning and scheduling resources of various technical disciplines and skill levels to multiple projects while achieving maximum potential life cycle value for the program. The process usually begins by development detailed requirements using customer needs and technology considerations as inputs. The product requirements are subsequently converted into specific tasks to be completed by individuals or teams on the project. The process of assigning tasks must take into consideration the type of skill required and the level of the skill required to complete the task in a timely manner. The process of assignment involves surveying the available resources and selecting the resource with appropriate skill type and level of expertise. Through this matching process the appropriate resources can be assigned to the tasks on the project.

The NPD project management problem structure is depicted in Figure 4.1. The following definitions are of primary importance to this dissertation:

1. Program – A program is a set of multiple NPD projects managed and coordinated by an enterprise/organization with the objective of achieving efficiencies of resources assignment across all projects.

2. Project – A project is a temporary endeavor consisting of tasks, resource assignments and schedules that is planned and managed to develop a new product or service. The organization has expectations that each project will result in a new product that achieves an objective value over a planned time horizon following completion of the project.

3. Requirement – A requirement is a condition or capability that must be met by the product to satisfy the standard, specification or imposed function desired by the customer. The requirements in total completely describe the new product and guide the objectives for the new product development project.

4. Task/Required Type/Durations/Precedence - Tasks are the primary component of work to be completed during the course of a project and in aggregate satisfy requirements and define the total scope of the project. Tasks have beginning and end dates that are constrained by duration estimates, logical precedence relationships and availability of resources that match the type requirements.

5. Potential Value/Life/Launch – Each project has a pre-specified potential revenue life cycle value that corresponds to a stationary market window with the time to potential launch equal to the time until the market opens and potential product life as ending when the market window closes.

6. Resource - Resources are the assets used to conduct the NPD project in the form of technical staffing or budget. This dissertation primarily deals with technical staffing (renewable human resources) and issues surrounding assignment of staff across multiple projects.

7. Resource Type – Each task has a requirement for a particular resource type or discipline such as mechanical engineering, electrical engineering, industrial engineering, etc. In general resources of different types are not interchangeable where task assignment is concerned.

8. Skill Level – Each resource is categorized by varying degree of competence or skill level which translates into differing levels of proficiency in completing tasks.

9. Work Rate – The work rate specifies the rate of completing a task which may vary based on the skill level (level of proficiency) of the resource assigned to the task.

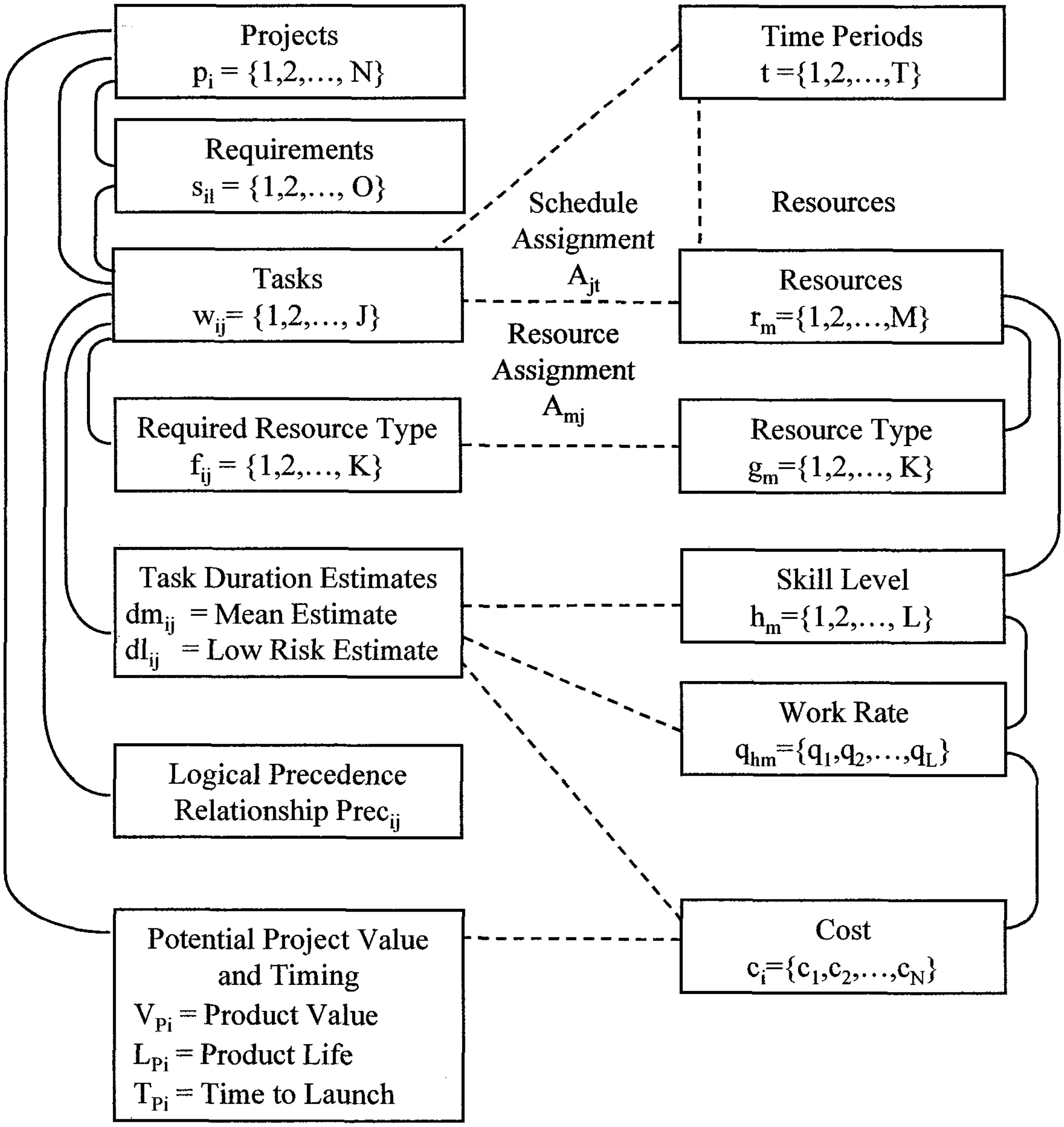

Projects
$p_i = \{1,2,\ldots, N\}$
Requirements
$s_{il} = \{1,2,\ldots, O\}$
Tasks
$w_{ij} = \{1,2,\ldots, J\}$
Required Resource Type
$f_{ij} = \{1,2,\ldots, K\}$
Task Duration Estimates
dm_{ij} = Mean Estimate
dl_{ij} = Low Risk Estimate
Logical Precedence Relationship $Prec_{ij}$
Potential Project Value and Timing
V_{Pi} = Product Value
L_{Pi} = Product Life
T_{Pi} = Time to Launch
Time Periods
$t = \{1,2,\ldots,T\}$
Schedule Assignment
A_{jt}
Resources
Resource Assignment
A_{mj}
Resources
$r_m = \{1,2,\ldots,M\}$
Resource Type
$g_m = \{1,2,\ldots, K\}$
Skill Level
$h_m = \{1,2,\ldots, L\}$
Work Rate
$q_{hm} = \{q_1,q_2,\ldots,q_L\}$
Cost
$c_i = \{c_1,c_2,\ldots,c_N\}$

Figure 4.1 Summary of the NPD Resource Assignment problem structure.

The variables shown in Figure 4.1 will be described in more detail later in this chapter.

The NPD project management problem solution requires the following decisions:

1. Establish the project priority for resource allocation.
2. Determine the resource priority.
3. Assign resources to individual tasks in order to achieve time-to-market financial objectives for the project/program.

Chapter 5 provides a focus on the project prioritization decision and describes the Robust Scheduling Method (RSM), the Project Valuation Method (PVM) and the Project Prioritization Method (PPM). Chapter 6 provides focus on resource assignment including description of the Resource Prioritization Method (RPM), the Constructive Scheduling Method (CSM) and the Skill Level Assignment Method (SLAM). Chapter 7 provides a description of the Resource Assignment for Multiple Project Scheduling (RAMPS) decision support tool. Chapter 7 also presents a series of computational experiments designed to test the model against simulated project and resource data.

4.2 Projects

The NPD resource assignment problem starts by defining a set of projects where the projects are numbered 1, 2,…,N.

$$p_i = \{p_1, p_2, \dots p_N\}$$

A project is defined (PMBOK 2008) as a temporary endeavor undertaken to create a unique product, service or result. New Product Development (NPD) projects are typically scoped and selected as part of the overall strategic plan for the business. The NPD portfolio selection process aligns product development projects with the strategic intent of the organization (Cooper 2001) and provides clear priorities to guide development activity. Planning initiatives result in opportunity statements, business cases, requirements, market driven project plans and high level schedules and targets for a set of NPD projects roughly sized to the capabilities of the organization. Once projects are selected at the strategic level the business plan is cascaded to the product development organization chartered with transforming the strategic plan into new products. As detailed project plans are formulated the development organization inherently faces significant challenges in achieving all strategic objectives due to very practical resource limitations and scheduling constraints.

4.2.1 Requirements

Each project consists of functional requirements that fully describe the product. A project requirement is a condition or capability that must be met by the product to satisfy the standard, specification or imposed function desired by the customer. Requirements include quantified and documented needs, wants and expectations of the customer or other stakeholders in the project. A set of O requirements is defined for each project.

$s_{il} = \{s_1, s_2, \ldots s_O\}$ = Requirement l of Project i

The process of establishing requirements for the NPD project typically occurs early in planning phase for the product. The process begins with the high-level customer functional requirements established in the Front End Process. High-level requirements are elaborated and parsed into a more detail level including the specification of what is to be developed. Depending on the extent and nature of the product the requirements may be simple or very complex. For simple projects, the project manager or engineering manager may assume responsibility for parsing and establishing more detailed requirements. In more complex projects, a system engineer may be involved with the complex task of breaking down requirements into a finer granularity. As shown in

Figure 4.1, the requirements may be grouped and organized with the intent to work towards more detailed specifications.

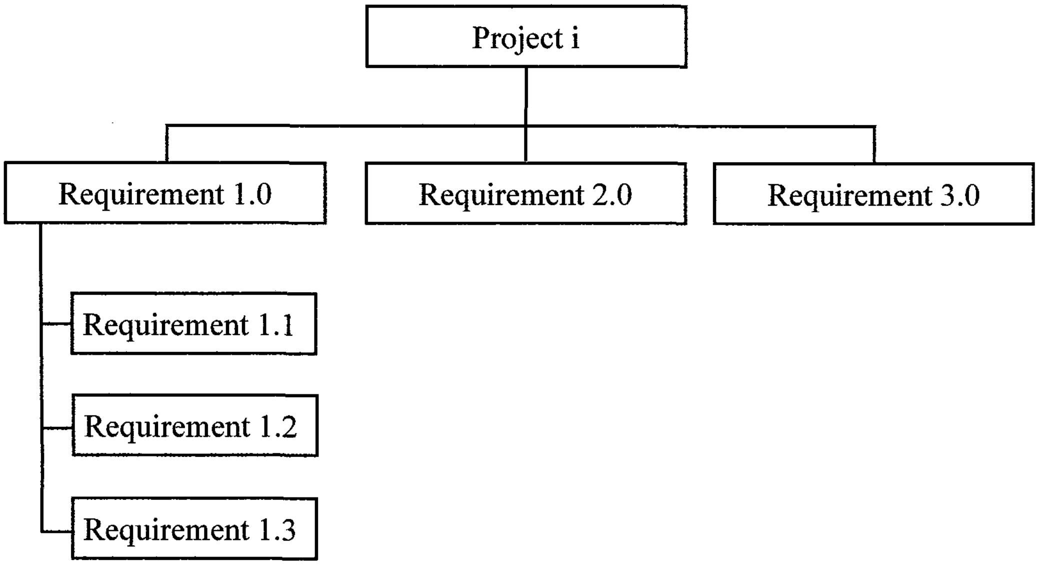


Figure 4.1 Project requirements progressively elaborated.

Requirements are established to achieve customer functional and performance objectives and to derive more detailed engineering specifications. The process of identifying requirements includes consideration for technical complexity and resolves requirements to the level they can be clearly understood by project management, technical management and the project team. The NPD project management formulation has the aim of resolving the requirements into specific work functions or tasks that can be assigned to the project team.

4.2.2 Tasks

The NPD project formulation uses the process of creating a Work Breakdown Structure (WBS) (Kerzner 2003) to map requirements to tasks in accordance with the way the work will be performed. The WBS is a deliverable-oriented hierarchical decomposition of the work to be executed and reflects the way in which tasks will eventually be organized and assigned to resources. Tasks are the primary component of work to be completed during the course of a project and aggregate define the total scope of the project. The task definition includes a description of deliverables, scope of work, duration of activities and related activities that may be completed before (predecessors) or after (successors).

The NPD process formulated in this dissertation defines tasks as work activities that correspond directly to each requirement. One task is defined for each requirement; thus a one-to-one correspondence of requirements to tasks. A set of J tasks is defined for each project.

$$w_{ij} = \{w_1, w_2, \ldots w_J\} = \text{Task } j \text{ of project } i$$

The simplifying one-to-one assumption enables simpler coding and computational efficiency in the assignment algorithms without loss of effectiveness of the solution. The one-to-one assumption can be relaxed in other formulations to have several requirements satisfied by one task or to define several tasks to complete a single requirement. The one-to-one corresponds of requirements to tasks is depicted in Figure 4.2

An additional assumption is that each task is assigned to one (1) resource. This assumption also simplifies the computational nature of the assignment process. This assumption precludes more than one resource from being assigned to a task. The one resource/task assumption may be relaxed in other formulations that assign tasks to several resources. For example it may be desirable to reduce the duration of a task and the project by assigning multiple resources to a single task. The aspect of multiple resources per task is not addressed in this dissertation.

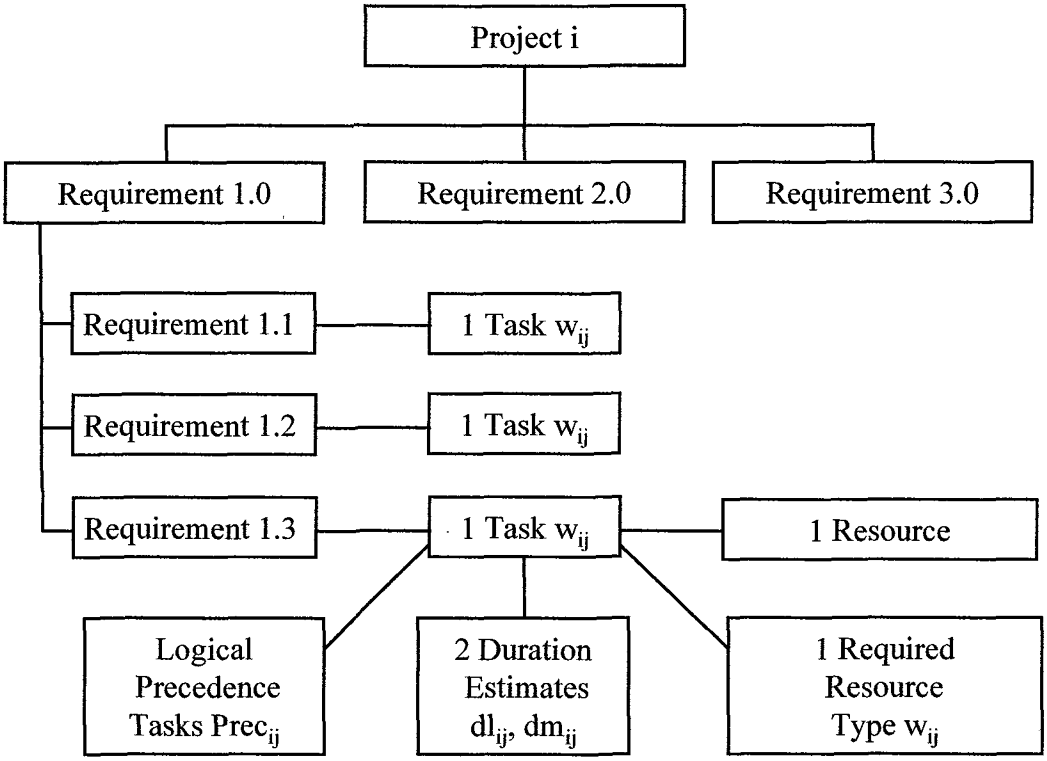


Figure 4.3 Requirements breakdown structure.

4.2.3 Required Resource Type

The NPD resource assignment formulation requires that the discipline or resource type be identified for each project task. NPD project typically require a variety of disciplines such as mechanical engineers, electrical engineers, software engineers, industrial engineers and test engineers to develop a new product. In addition the same NPD project may also engage manufacturing engineers and technicians. In this dissertation the requirements are parsed at a sufficient level of detail such that each task has one resource type as depicted in Figure 4.2. Thus a one-to-one correspondence exists between requirements, tasks, resource and resource type. Each task is formulated with one of K resource types.

$f_{ij} = \{f_1, f_2, \ldots f_K\}$ = Required Resource Type

One area of particular interest is process associated with skill assessment including discipline, specialty and skill level. This step is often taken for granted in project planning. On the surface this process seems to be straightforward. Simply identify what needs to get done and find someone competent to do it. Managers approaching the process in this simplistic way risk underestimating the complexity of the project tasks and risk overestimating the skills and capabilities available to complete the tasks.

4.2.4 Task Duration Estimates

Traditional project management techniques including the Performance Evaluation and Review Technique (PERT) and the Critical Path Method (CPM) form the basis for organizing and planning new product development projects. The traditional algorithms are derived from networks of sequential tasks each having varying start and end dates and accounting for interdependencies between tasks. The critical path approach is used to determine the longest network of dependent tasks which defines the overall length of the development from which the launch date can be derived. The PERT technique may be used to include statistical variation (optimistic, mean and pessimistic) for task durations and for the project estimates.

Project management is made even more robust through statistical methods embodied in Critical Chain (CC) Project Management method (Goldratt 1997). The CC method is based on the Theory of Constraints and accounts for both logical network sequencing as well as resource constraints in the project planning process. In CC project planning two estimates are formulated for task completion including a mean estimate time dm_{ij} and a low risk estimate of completion time dl_{ij}.

dm_{ij} = Mean duration estimate for task j project i.

dl_{ij} = Low risk duration estimate for task j project i.

The mean estimate corresponds to a 50% confidence that the task will be completed by $dm_{ij.}$ A low risk estimate corresponds to a 90% confidence that the task will be completed by dl_{ij}. The use of two estimates simplifies the process of estimation as compared to traditional PERT methods that require three estimates (e.g. optimistic, mean and pessimistic).

The right skewed lognormal distribution shown in Figure 4.3 is used (Tukel 2006) to represent the task duration. The two estimates dm_{ij} and dl_{ij} may be used to compute the uncertainty in the project task duration estimate. The uncertainty for the lognormal distribution has been shown (Herroelen 2001) to be a multiple in the range of 0.05-to-1.5 times the standard deviation. To facilitate simplicity in computations in this research the uncertainty is assumed to be a multiple of one times the standard deviation. Thus the difference between dl_{ij} and dm_{ij} represents the estimate for standard deviation for the distribution.

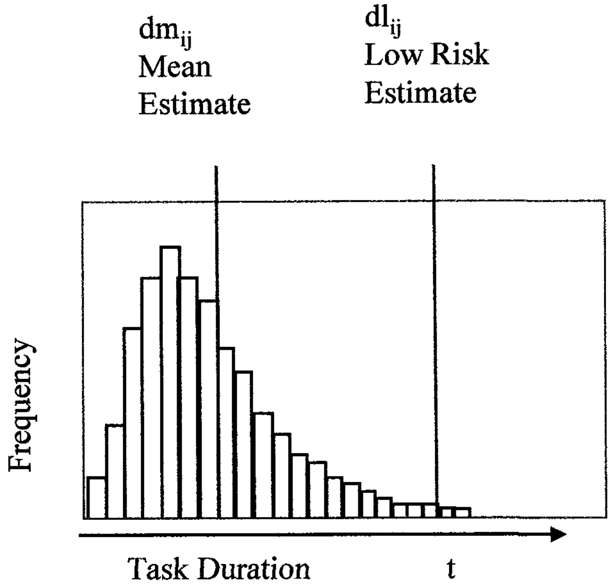


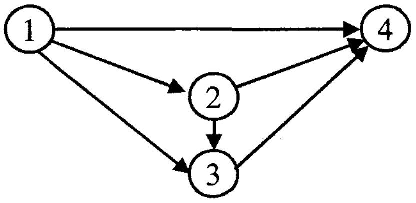
Figure 4.3 Project task durations estimates shown as a lognormal distribution.

4.2.5 Logical Precedence Relationships

The NPD resource assignment formulation requires specification of the logical network of tasks to reflect sequence in which activities are to be performed. The logic network specifies when a task can begin based on predecessor tasks that must be completed before the task is started. The precedence relationship allows the use of traditional critical path algorithms to be applied to determine the Critical Path (CP) and critical path tasks (CPT). The precedence logic should be included in the planning as early as possible for construction of schedules used in prioritization. Each task is identified in the definition of the project network along with a set of precedence tasks that must be completed before the task starts.

$Prec_{ij}$ = Tasks that must be completed on project i before task j can start.

The precedence relationship is typically reflected in matrix form to represent both the predecessor tasks and successor tasks for the entire project. The successor task relationship is required to determine slack in the project network. The logical relationship indicator is provided by a zero/one (0/1) entry in the matrix. A simple example of the predecessor (and successor) matrix is illustrated in Figure 4.4.



$$\text{Prec}_i = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4.4 Example of the precedence matrix Prec for a simple project.

In this example the columns in the matrix represent each task numbered one (1) through four (4). In the example column one represents predecessors for task one and since task one has no predecessors all entries are zero. Column two representing predecessors for task two has only one entry in row one indicating that task one is a predecessor two. Column three representing predecessors for task three has two entries, row one and row two indicating that task one and two are predecessors of task three. Column four representing predecessors for task four has three entries, row one, row two and row three indicating that tasks one, two and three are all predecessors of task four.

The same Prec matrix can be used as a successor matrix by allowing the rows to indicate successor tasks. For example row one representing the successors of task one has three entries, column two, column three and column four to indicate that tasks two, three and four are successors of task one. Similarly all rows reflect the successor tasks of the task corresponding to the row number.

4.2.6 Potential Value and Market Timing

The NPD resource assignment formulation requires knowledge of the potential project value, time-to-market and product life cycle. These parameters are formulated in the planning stage as part of the financial analysis. The planning initiatives result in a business case with forecasts of sales and profit potential. The business case also highlights the market driven launch date required. The launch date corresponds to the market window opening date or the earliest date the product can be sold. The business case forecast also provides direction on the life cycle for the product and the planned end date. The end date corresponds to the market close date or the last date the product can be sold. This strategic plan provides the potential market value for each new product and identifies the launch date that must be achieved in order to capture the potential life and value for the product. The following parameters are illustrated in Figure 4.5.

V_P = Potential product value (revenues) over the product life cycle.

T_P = Potential time to launch at earliest date product can be sold.

L_P = Potential product life cycle.

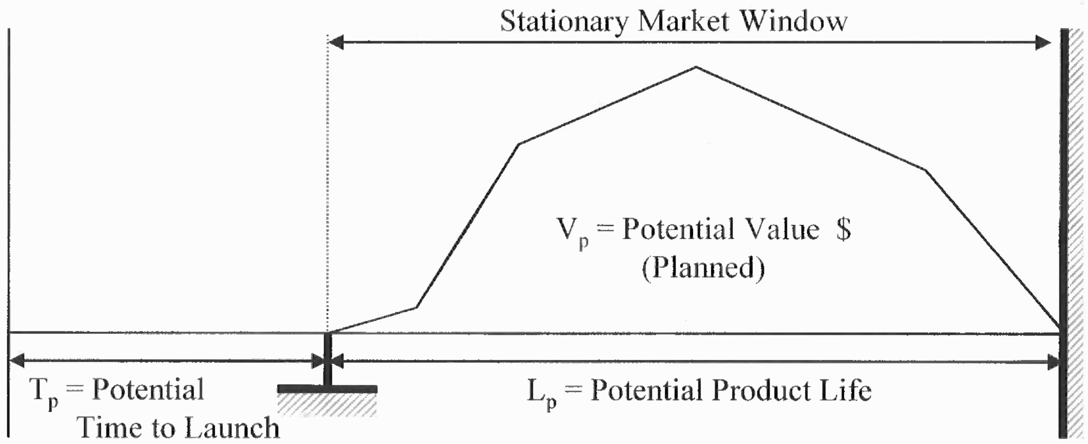


Figure 4.5 NPD potential value, potential product life and time to launch.

4.3 Resources

Resources are the assets used to conduct the NPD project in the form of technical staffing or cash budget. This dissertation primarily deals with technical staffing (renewable resources) and issues surrounding assignment of staff across multiple projects. To a lesser emphasis the project costs (budget or non-renewable resources) is tracked for future reference. NPD resource assessment involves understanding the capabilities of the available technical professionals who will carry out the work on the NPD project. An initiative is often undertaken to characterize the resources and produce an inventory defining the exact disciplines and capabilities available for assignment. The resource inventory identifies the resource quantity, type (e.g. engineering discipline), skill level and work rate. This formulation defines a set of M resources for assignment to NPD projects.

$r_m = \{r_1, r_2, \dots, r_M)$ = Set of resources available for assignment.

The objective of the skills inventory is to use the data to make efficient and effective project task assignments. The project manager and functional manager can refer to a summary skills inventory report on an ongoing basis to select appropriate staff members for projects. The inventory may be updated periodically to include new staff members and to adjust levels based on new skills that are acquired by existing staff members.

4.3.1 Resource Type

Depending on the extent and nature of the NPD project, the development may require a variety of professions with differentiated technical skills. For example development of a new telecommunications product development may require systems engineers, electrical engineers, mechanical engineers, software engineers, manufacturing engineers and a variety of technicians. Each technical discipline category may involve work of different complexity and different levels. The NPD formulation defines a set of K resource types.

$g_m = \{g_1, g_2, \ldots, g_K) =$ Set of resources types

Resource skill types inventoried at this level are generally not interchangeable in the short term. For instance you can not interchange an electrical engineer who specializes in circuit design with a software engineer who does coding for protocol interfaces. While it is possible that an extraordinary individual could jump back and forth it is generally accepted practice that these resources must specialize to achieve the efficiencies required on NPD projects.

4.3.2 Skill Level

To achieve efficiency the skill level definition typically requires greater resolution beyond the particular discipline. The organization may have technical professions with various levels of training and with some operating at a more senior level than others. Each of the staff resources having a particular discipline mentioned here also have varied experiences and often specialties that make them efficient in some tasks and inefficient in others. Defining the skill level will become important as we follow the process to task assignment.

The primary responsibility of the management is to ensure the organization has access to skilled resources of the types appropriate to develop the scope of products within the organizations charter. All skill types reflect technical professionals with varying degree of skills needed to complete tasks assigned under the project plan. Differences between the levels of skill among technical professions may be significant and could profoundly alter the project productivity and impact the overall schedule. For instance one staff member may have the skill level required to a greater extent than his peers which will ultimately allow that individual to be much more efficient in completing a particular task. Individuals containing only a moderate level of skill necessary may complete the task but the expected duration may be much long than the individual who is more proficient. Finally organizations may include individuals with a lesser skill level for some tasks but not for others. Placing individuals with lesser skills onto more complex tasks or tasks of greater magnitude will likely lead to productivity and schedule impact.

In this formulation we identify the skill levels for each skilled professional as a precursor to the task assignment process. For skill level to be useful in the planning process the characterization must be simplified. The characterization proposed in this dissertation is a simple numeric that defines three levels (or grades) of resource.

> Level 1 - Possess required skills to a greater extent and likely to complete assigned tasks faster than average; for example a senior engineer.
>
> Level 2 – Possess required skills to complete assigned task in an average amount of time; for example a full experienced engineer.
>
> Level 3 – Possess required skills to a lesser extent and likely to have difficulty completing tasks in an average amount of allotted time; for example a junior engineer or technician.

A set of three (3) skill levels is defined (L=3).

$h_m = \{h_1, h_2, \ldots , h_L)$ = Set of skill levels, where L=3 in this formulation.

It is important to note that this approach is not meant to grade individuals for the purposes of salary leveling or for performance compensation. Enlightened management recognizes that every population has a range of individuals available and that no two individuals have exactly the same skill levels.

Many organizations succeed and thrive without attention to skill level assessment where task assignments are based on a prior knowledge of the capabilities of individuals. These organizations may assign resources of similar skill level from one project to the next subsequent project without considering skills. Another example of where skill assessment may play a lesser role is an organization that has a base of experienced staff and low turnover such that established skill types and skill levels are understood by all.

4.3.3 Work Rate Transformation

In general resources of different skill levels are interchangeable but unplanned substitutions risk an impact to productivity and project schedules. To achieve efficiency the skill level is accompanied by a work rate that defines the relative speed that a resource can complete a task. In this formulation we define a one-to-one correspondence between skill levels and work rate. For example a level one (1) resource has one rate, a level two (2) has a second rate and a level three (3) has a third rate. The work rates are all defined relative to level two (2) which is the average rate work can be achieved. The work rates parameter is defined as follows:

$q_h = \{q_1, q_2, \dots, q_L)$ = Set of skill levels, where L=3 in this formulation.

$$0 \leq q_1 \leq 1$$
$$q_2 = 1$$
$$0 \leq q_3 \leq 1$$

Work rate differences will result in transformation of task durations. This transformation will become important in resource skill level assignment (Chapter 6) where substitution of resources between projects and tasks is considered to improve efficiency. The transformation for task durations is based on a unit value for level two (2), a decreased duration for level one (1) and an increased duration for level three (3).

$$dm_{h=1} = dm_{ij}\,(1 - q1)$$
$$dl_{h=1} = dm_{ij}\,(1 - q1)$$

$$dm_{h=3} = dm_{ij}\,/(1 - q3)$$
$$dl_{h=3} = dm_{ij}\,/(1 - q3)$$

4.4 Assignment Variables

This section defines the variables for assigning tasks and schedules. The methods for resource assignment and scheduling tasks are covered in more depth in Chapter 6.

4.4.1 Task Assignment

The task assignment process match tasks to resources on projects. The assignment variable A takes on a value of m if resource m is assigned to task j and is assigned and a value of zero (0) if unassigned. This resource assignment activity accounts for task duration and ensures that resources are available to complete the entire task from start to finish. In order for a resource to be assigned it will need to match resource type requirements and be available for the duration of the task.

$$A_{mj} = \begin{cases} 0 & \text{if task j is not assigned} \\ m & \text{if resource m is assigned to task j} \end{cases}$$

In working environments it is important to carefully make assignments to avoid mistakes of the wrong discipline or the wrong level of proficiency. Problems are likely to occur in fast-paced environments and where managers have an incomplete knowledge of the professional technical disciplines and skill levels. Frequently assignment involves an interactive approach in collaboration with resources actually doing the work to increase the likelihood of proper matching. The process of task assignment can be facilitated by improved methods of resource and skill inventory that make clear the skill types and levels available for assignment and allow the project planning process to be completed on an expedited basis.

4.4.2 Time Period Assignment

In a larger scope matching skilled resources to tasks needs to consider the time factors since every project operates within time constraints. The projects in this formulation occur within T time periods where each time period is defined as one day.

$t = \{t_1, t_2, \ldots, t_T)$ = Set of time periods for implementing the program.

Once duration estimates and precedence relationships are formulated schedules can be developed for assignment. The objective of the scheduling effort is to define the start and end dates for each task and for the project in entirety.

ST_{ij} = Start date for task j of project i

FN_{ij} = Finish date for task j of project i.

The start and finish dates for tasks may be progressively elaborated as schedules are formulated, refined and evolve during the course of resource assignment. The final schedule will account for each task and the resource assignments by time period over the course of the project. The assignment variable A takes on a value of m if resource m is assigned to task j during period t and is assigned and a value of zero (0) if unassigned.

$$A_{mjt} = \begin{cases} 0 & \text{if task j is not assigned} \\ m & \text{if resource m is assigned to task j in period t} \end{cases}$$

4.4.3 Skill Level Assignment

The improvement heuristics developed as part of the Skill Level Assignment Method (SLAM) require the assignment of skill levels to tasks in order to take advantage of the work rate differential associated with higher/lower skill level resources assigned to the task. As alternate trial solutions are produced the task must maintain the assigned skill level resource or alternatively be transformed by a substitution. Only when a substitution is accepted to improve productivity will the task skill level assignment be changed. The assignment of a skill level to a task is indicated by:

$$A_{hj} = \begin{cases} 0 & \text{if task j is not assigned} \\ h & \text{if skill level h is assigned to task j} \end{cases}$$

Robust scheduling methods are covered in detail in Chapter 5 and resource assignment methods are covered in Chapter 6.

4.5 Cost and Budget Considerations

NPD cost and budget management is an important topic since each formulation of each resource assignment and scheduling approach may result in a different cost. It may be of interest to understand the cost from the perspective of investment and Return On Investment (ROI) for new products. NPD costs also may be of interest to understand how the performance of the model can be improved through additional allocations to the budget. The cost in this formulation is closely aligned with the resource inventory and incremented as resources are added to the inventory. The cost for resources of different resource types and skill levels are accumulated to tracked to identify the cost of a particular resource set. While NPD cost management and product cost targeting can easily be tracked in this formulation the cost management aspect of NPD project management is de-emphasized in this research.

Alternatively this research assumes a fixed budget as the basis for resource assignment. The fixed budget is representative of actual industrial settings where NPD budgets are closely aligned to the annual operating budget for the firm. The annual operating budget is typically established six to twelve months in advance of the annual period NPD activities. Thus the annual budget defines within some tolerance the number of staff (head count) and support resources allocated to the NPD activity for ongoing planning. When viewed from the annual budget perspective the NPD resource assignment problem is essentially a net sum zero activity regarding cost. The NPD organization is chartered with completing the projects and spending the budget. The challenge within that charter is to deploy resources to achieve the market windows and launch dates that enable financial returns for the firm.

It may be argued that NPD costs are useful in comparing one project vs. another for potential assignment and allocation of resources. However where one project tends to dominant over another based on cost or returns the priority decisions and resource allocations tend to be obvious. The research in this dissertation considers projects of similar size and scope where the cost will usually be similar.

CHAPTER 5

PROJECT PRIORITIZATION

Chapter 5 and Chapter 6 satisfy Research Objective No. 3 by developing a decision model for SLTI NPD resource assignment in a multi-project environment. This chapter includes a time sensitive project value model that is tightly coupled to the opportune market launch date and progressively penalizes product values with schedules that are delayed. The model includes decision parameters, variables, constraints and an objective function that strives to optimize the value of the project portfolio by minimizing the value lost as a result of NPD schedule delays.

5.1 Prioritization Approach

This research introduces a time-value model for establishing scheduling priorities at the tactical level to assist development organization in assigning resources to SLTI NPD projects. The model considers the unique complexity of each project and uses Critical Chain (CC) project scheduling techniques (Goldratt 1997) to develop task duration estimates and robust baseline project schedules. The model links schedules to a valuation model that simulates the stationary market window over the life-cycle of the product. Inherent in the model is the assumption that any delay in time-to-market results in decreasing value to the project. The time-value linkage forms the basis for establishing scheduling priorities on a current and consistent basis. The prioritization methodology serves as a starting point for resource assignment in multi-project environments where time-to-market is a primary success factor.

The approach to prioritization follows the process outlined in Figure 5.1.

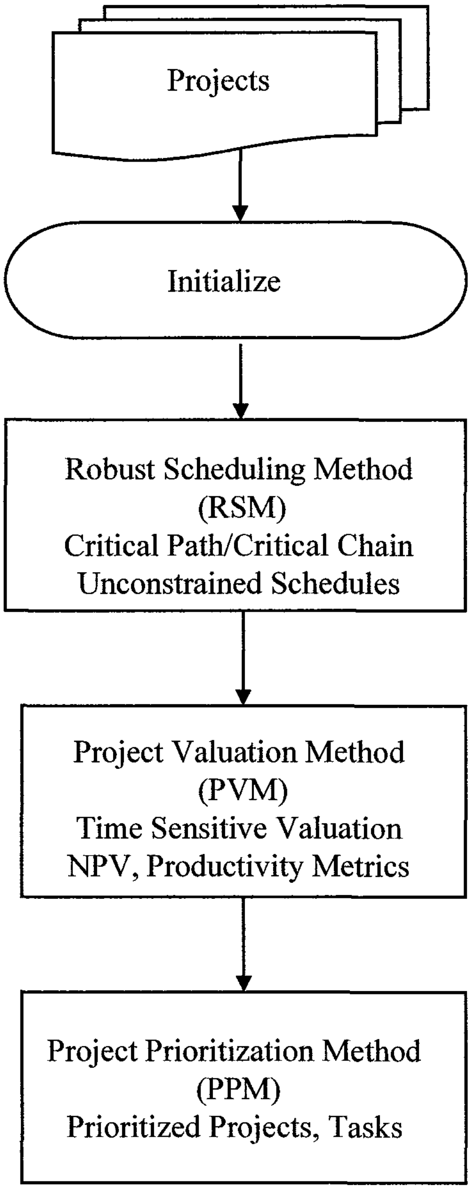

Projects
Initialize
Robust Scheduling Method (RSM)
Critical Path/Critical Chain
Unconstrained Schedules
Project Valuation Method (PVM)
Time Sensitive Valuation
NPV, Productivity Metrics
Project Prioritization Method (PPM)
Prioritized Projects, Tasks

Figure 5.1 Process for NPD project prioritization.

5.2 Project Portfolio Selection

Classical operations research techniques such as Integer Programming, Real Options Theory, Analytical Hierarchy Process, all offer potential to improve the process for selection and screening NPD projects. In addition scoring models offer significant advantage in addressing qualitative and strategic factors in prioritization. More often in practical settings the more academic approaches are supplanted by traditional financial techniques that are easier to apply. Financial methods tend to provide clear direction on priorities when one project has sufficiently larger value and dominates over other projects by comparison. Where projects are closer in value, more careful financial analysis may be employed such as expected commercial value, net present value, bang-for-the-buck and productivity index (Cooper 2001). The productivity index computed as the ratio of potential value (e.g. NPV) of the project to resources consumed to complete the project is emphasized in this research.

$$\text{PI} = \text{Productivity Index} = \frac{\text{Value}}{\text{Resources Consumed}}$$

Using a technique such as the productivity index method all projects are assigned a priority based on a common metric and arranged in priority order from highest to lowest value of the productivity. The output of the strategic plan provides a set of prioritized projects (2) roughly sized to the capacity of the development organization.

$$P_i = \{p_1, p_2, \cdots p_N\}$$

The prioritization method may be augmented to include risk factors such as the probability of technical success or probability of commercial success to reflect the confidence associated with product development and commercialization each project. In addition the productivity index may subtract the development expense and include product cost and the net present value computation to arrive at a consistent economic value for comparison across projects. The result of the productivity index method is the ranking and prioritization of NPD projects consistently across projects.

5.3 Robust Scheduling Method (RSM)

Traditional project management techniques including the Performance Evaluation and Review Technique (PERT) and the Critical Path Method (CPM) form the basis for organizing and planning new product development projects. The traditional algorithms are derived from networks of sequential tasks each having varying start and end dates and with intricate interdependencies between tasks. The critical path approach is used to determine the longest network of dependent tasks which defines the overall length of the development from which the launch date can be derived. The PERT technique may be used to include statistical variation (optimistic, mean and pessimistic) for task durations and for the project estimates. Project management is made even more robust through statistical methods embodied in Critical Chain (CC) Project Management method (Goldratt 1997). The CC method is based on the Theory of Constraints and accounts for both logical network sequencing as well as resource constraints in the project planning process.

5.3.1 Task Estimatation and Tolerance

The critical chain project planning method typically draws on the lognormal distribution (Tukel 2006) shown in Figure 4.3 to represent the task duration estimate. The lognormal distribution is a right skewed distribution that reflects some of the characteristics of how work activities are actually are carried out. For example Parkinson's Law states that work tends to expand to fit the available time which is reflected by the long tail to the right of the lognormal distribution. The critical chain project planning technique typically employs two estimates for task duration including a mean estimate dm_j and a low risk estimate dl_j. The mean estimate corresponds to a 50% confidence that the task will be completed by dm_{ij}. A low risk estimate corresponds to a 90% confidence that the task will be completed by dl_{ij}. The use of two estimates simplifies the actual process of estimation as compared to traditional PERT methods that require three estimates (e.g. optimistic, mean and pessimistic).

The uncertainty (tolerance between dl_{ij} and dm_{ij}) is a multiple of m times the standard deviation. The multiple m for the lognormal distribution has been shown (Herroelen 2001) to be in the range of 0.05 to 1.5. This dissertation makes a simplifying assumption that the multiple is equal to one (1) thus the estimate for standard deviation may be computed as the difference between the low risk estimate and mean estimate.

$$sd_{ij} = dl_{ij} - dm_{ij}$$

The set of logical precedence tasks for each task is identified in the definition of the project network is be represented by $Prec_{ij}$ {set of precedence tasks for task j}. Traditional critical path algorithms are applied including forward pass to determine the

Critical Path (CP) defined as the longest chain of precedence dependent tasks and backward pass to determine the set of critical path tasks (CPT). The portfolio selection process has been completed by assuming that the projects identified are sized to the capacity of the organization and that baseline schedules are resourced and leveled to the set of projects outlined in the strategic plan.

5.3.2 Critical Path Method (CPM)

The robust scheduling method employs a standard critical path method (CPM) algorithm for determining scheduling parameters. The method assumes a formulation (as in Chapter 3) where task durations (dm_{ij}) and precedence ($Prec_{ij}$) relationship ships are know. The method assumes a right directed Activity on Node (AoN) network where tasks are numbered such that predecessors always have a lower number and successors have a higher number. If the network is not constructed with a null node at the beginning and end a start and end node of zero duration is added to facilitate the computations. The path of longest duration through the project network is called the Critical Path (CP) since it determines the minimal project duration.

The CPM algorithm uses a forward pass to determine the Early Start (ES) date and the Early Finish (EF) date for each task. The forward pass starts by assuming ES and EF for the first node are zero and progresses to the right in ascending node order. ES for a successor node j is the maximum of the EF times of the proceeding nodes. EF for node j equals the early start of node j plus the duration of j ($ES_j + dm_j$).

The backward pass is used to determine the Late Start (LS) date and the Late Finish (LF) date for each task. The backward pass typically starts by assuming that the

LF date for the last node is equal to the EF date for the same node. This assumption is true for networks where minimum slack is equal to zero and where no delays in the final schedule are allowed. The backward pass progresses from the last node through the network backwards in descending node order. LF for a predecessor node j is the minimum of the LS times of the immediate predecessors. LS for node j equals the late finish of node j minus the duration of j (LF_j - dm_j).

The Total Slack (TS) for each task is then easily given as LF_j – EF (or equivalently, as LS_j-ES_j). This is the amount of time an activity may be delayed beyond its early start time without causing a delay in the time of the expected launch date. The Free Slack (FS) is the amount of time an activity may be delayed from the ES without delaying the start of any of it's successors. Free slack for activity j is computed as the minimum ES of it's successors minus EF_j. Activities with zero slack are called critical activities, and are always found on one or more critical paths. Critical path(s) determine minimum project duration since all of them must be completed as part of the project, and all non-critical activities can be completed concurrently with the critical activities. The mean estimates from each task are used to derive the mean estimate for the project duration; i.e. the critical path (CP) (mean project duration) is defined as the sum of the means of the tasks on the critical path where critical path tasks have been identified through minimum slack criteria.

$$CP_i = \sum dm_{ij} \qquad j \in \text{ Critical Path Tasks}$$

5.3.3 Critical Chain Buffer Management (CCBM)

Recent literature suggests (Lechler 2005) that key aspects of the CC methodology may be applied beneficially without implementing the entire CC approach. For instance it may be practicable for project managers to focus on just the project buffer than to monitor intricate layers of feeder and resource buffers. The approach of focusing primarily on the project buffer has been called Critical Chain Lite (Lechler 2005) of Simplified Critical Chain method (Leach 2003-1 The simplified approach is carried forward in this research and used to establish baseline schedules that incorporate associated variation. The schedule estimate is developed through combination of the critical path mean duration estimates and a CC project buffer. Thus it is recommended (Leach 2003) that the project buffer be sized with a combination of the project Standard Deviation (SD) estimate and an additional Bias Buffer (BB).

$$\text{Project Buffer (PB)} = \text{SD} + \text{BB}$$

The project Standard Deviation (SD) is computed using the square root of the sum of the squares (SSQ) of the standard deviations of tasks on the critical path

$$\text{SD}_i = \sqrt{\sum_j \text{sd}_{ij}^2} \qquad j \in \text{ Critical Path Tasks}$$

Research on project buffer sizing (Geekie 2008) shows that the SD buffer size is dependent on the number of tasks and does not adequately capture the variation associated with longer projects, higher numbers of merge points or the tendency in underestimated task estimates. Thus an additional factor is included in the project buffer to account for estimate bias.

The literature (Leach 2003-2) suggest that the total project buffer should be at least 25% of the mean path duration and that a project buffer sized at 50% of the mean path duration is conservative for most all projects. Thus project managers should strive to achieve overall project buffers between 25% and 50% of the mean path duration. In the absence of control charts a parameter ρ is introduced in to establish a total project buffer that is greater than 25% of the mean path duration.

$$\rho \geq 0.25 \; - \left(\frac{SD_i}{CP_i} \right)_{\min \in i} \qquad 0 \leq \rho \; \geq 0.5$$

Once ρ that is selected the Bias Buffer (BB) is computed as a multiple of the mean path duration.

$$BB_i = \rho \, CP_i$$

The expected project duration is computed as the sum of mean critical path duration, project standard deviation and bias buffer.

$$T_{Ei} = CP_i \; + SD_i + BB_i$$

The project depicted in Figure 5.2 shows an example of five critical tasks where each critical task is started at the mean estimate of the proceeding critical task. Dedicated resources are assigned to complete tasks in "road runner" fashion with planned variation around the mean estimate of each task. The project buffer is a combination of the variation buffer and the bias buffer. The project buffer is included and managed at the end of the project to form the robust project schedule.

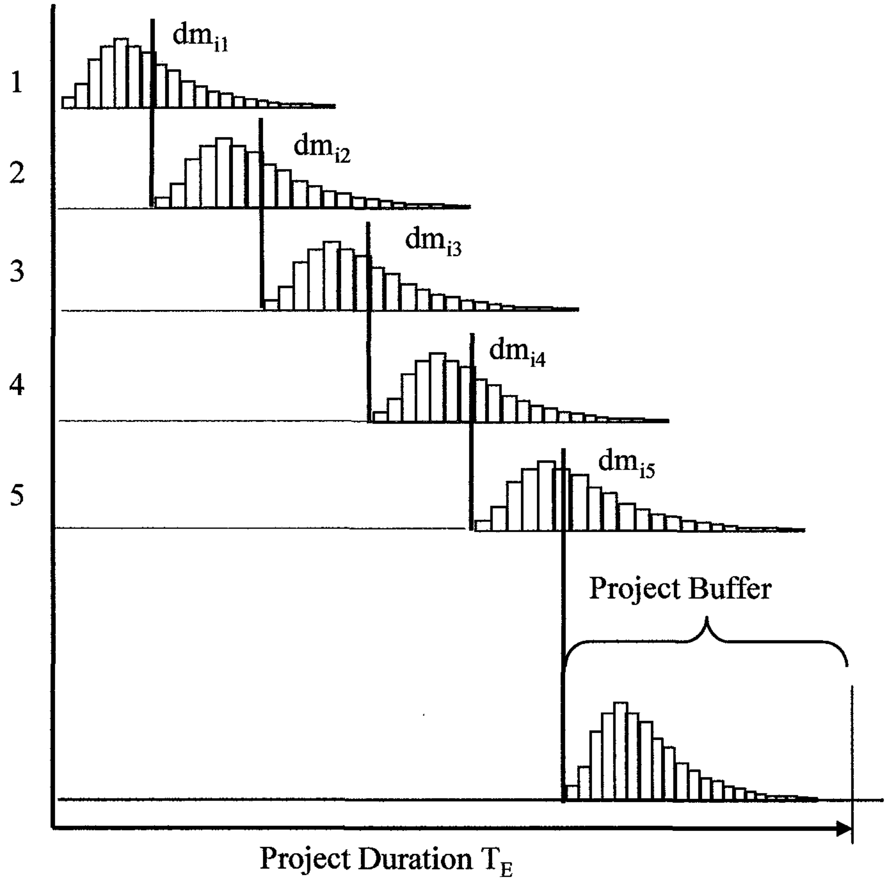


Figure 5.3 Robust project scheduling.

Accommodating estimate variation is a critical aspect developing a robust baseline schedule. Ultimately this variation may impact the ability of the organization to deliver the product on time and to achieve financial goals for the NPD project. The scheduling approach described here is vastly simplified and abbreviated for the purpose of establishing a baseline schedule as input for the prioritization process. However if sufficient detail is available a full blown Critical Chain resource leveled schedule may be developed to include resources buffers, feeder buffers and capacity buffers to accommodate multi-project planning environments.

5.4 Project Valuation Method (PVM)

NPD projects have the primary objective to create value and generate revenue/profit for the firm. The business case is used to forecast sales revenues and margin over the planned life cycle of the product.

5.4.1 Triangular Product Life Cycle

The life cycle progression for a typical SLTI product is modeled in Figure 5.4 and includes progression through product introduction, growth, maturity, saturation and decline.

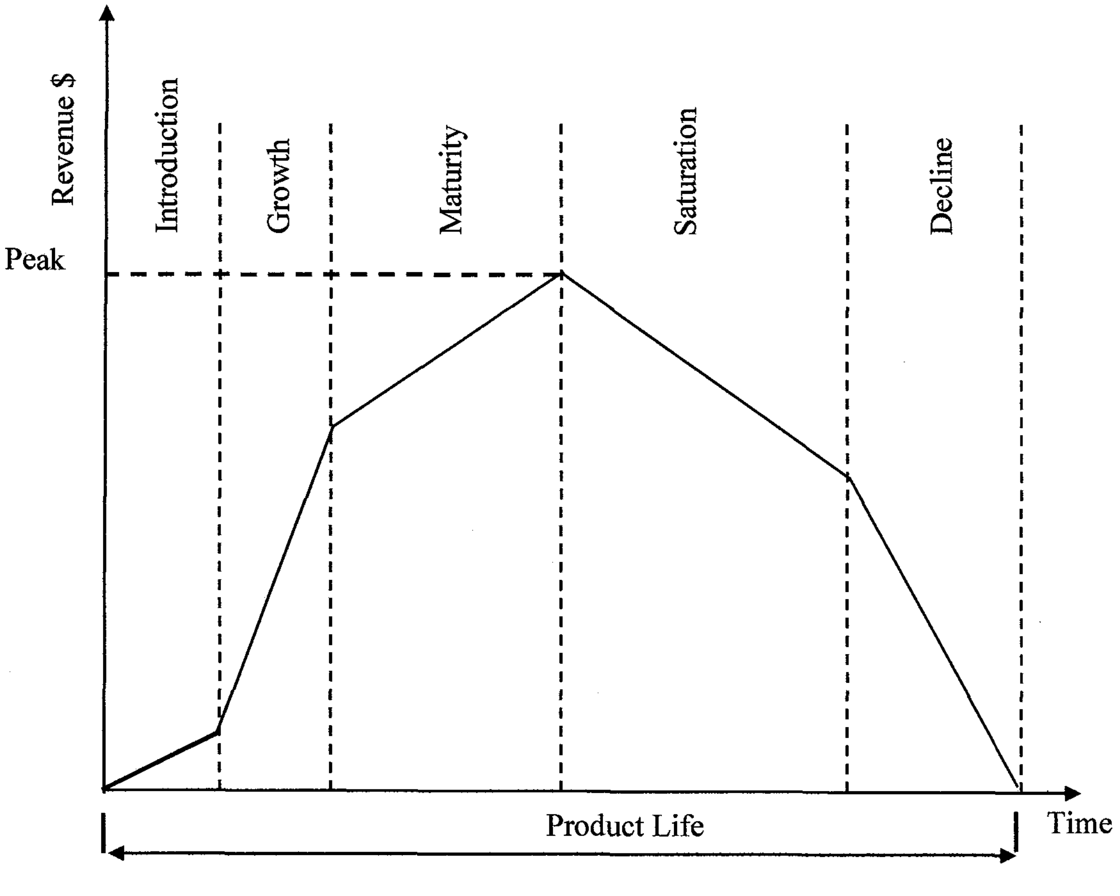


Figure 5.4 Product Life Cycle trend.

Source: Sassenburg, H., Berghout, E. (2005) "The NPVI Method to Support Market Entry Strategies for Sofware Products."

In the introduction stage sales tend to be low as new product is just being introduced and in the growth stage sales tend to grow rapidly and the profits rise. In the maturity stage the sales peak or level off and then begin to decline in the saturation phase where multiple competitors have entered the market. As the competition becomes even more energetic life cycle sales decline until the end of the product life. The product life cycle is approximated with a triangular value function as shown in Figure 5.5 (Liu 1996, Sassenburg 2005) which implies a constant linear growth and decline.

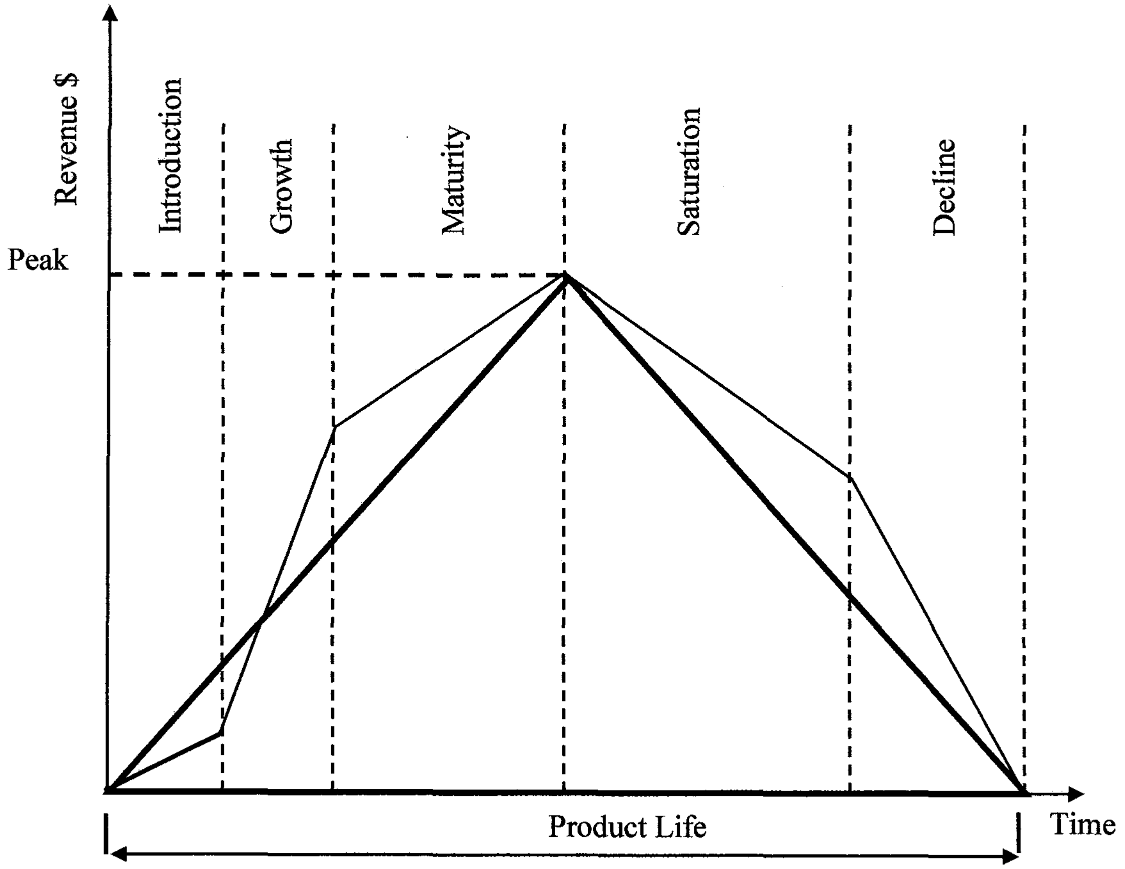


Figure 5.5 Product Life Cycle (PLC) with approximate Triangular Value.

Source: Sassenburg, H., Berghout, E. (2005) "The NPVI Method to Support Market Entry Strategies for Sofware Products."

5.4.2 Time-Value Model

The triangular geometry is a useful in computing the project value represented as the area under the triangular curve. The potential project value is given by V_p where the product life is L_P and the peak revenue is R_P.

$$V_P = \frac{1}{2} L_P \times R_P$$

The potential financial objective assumes that the product will be launched at the optimal point in time necessary to achieve desired market share and maximum potential revenue V_P over the product life cycle. Projects often experience delays relative to potential product launch dates due to resource and scheduling limitations. Since the window of opportunity for selling the product is stationary, if the expected project schedule T_E exceeds the potential schedule T_P then a delay is introduced and the expected life is compressed. The triangular life-cycle model (Liu 1995) assumes that the peak of the products value occurs in the middle of the life-cycle and is independent of when the product is actually launched. The delay D compresses the life cycle of the product and results in the smaller triangle representing the expected value V_E. The triangular models for V_P and V_E have the same rate of increase and therefore the model is defined in the region $\leq L_p/2$. The lost value V_L is the difference between the areas of the two triangles as shown in Figure 5.6.

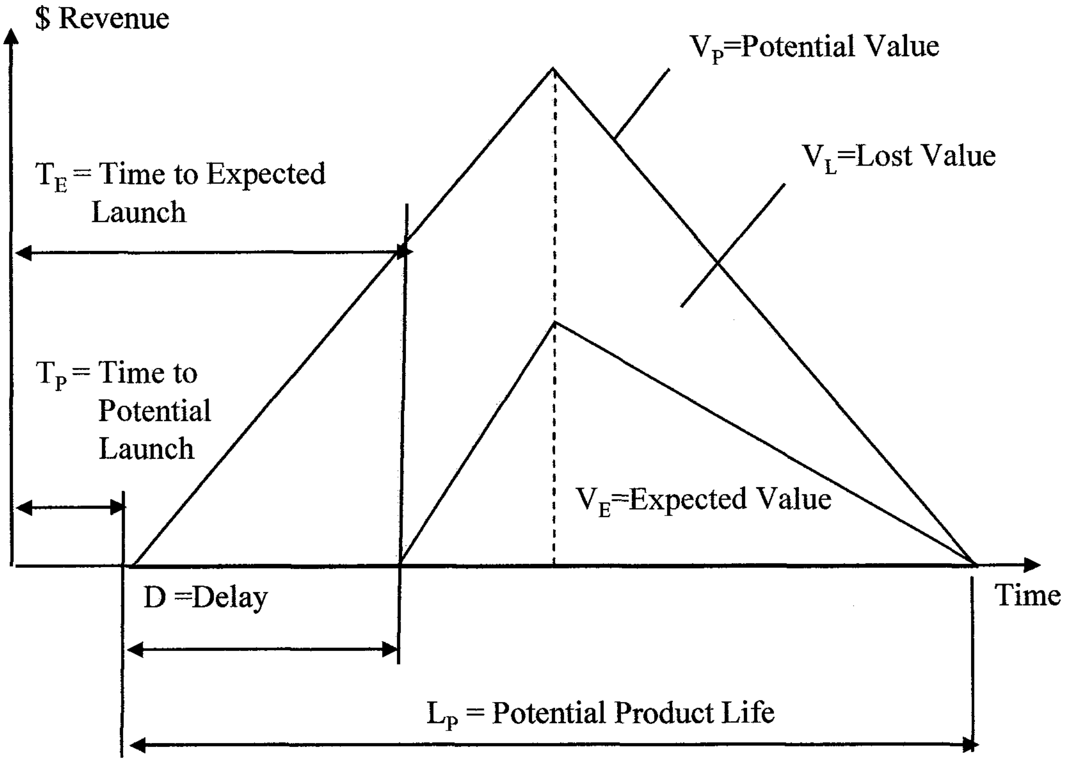

$ Revenue
V_P=Potential Value
V_L=Lost Value
T_E = Time to Expected Launch
T_P = Time to Potential Launch
V_E=Expected Value
D =Delay
Time
L_P = Potential Product Life

Figure 5.6 Triangular Product Life Cycle model.

Liu, J. (1995) "Detailed model shows FPGAs' true costs." Electronic Design News, May 11, 1995. pp. 153-156.

The triangular model provides a simplified decomposition of the potential value as the sum of expected value and lost value.

$$V_P = V_E + V_L$$

In addition the delay is defined as the difference between the expected time to launch and the potential time to launch.

$$D = T_E - T_P$$

The triangular model provides a convenient construct for relating expected value and lost value lost as a function of the potential value V_p, the potential product life L_p and the time t (or delay D).

$$V_E = V_P\left(1 - \frac{3}{L_P}t + \frac{2}{L_P^2}t^2\right) \qquad \text{for} \quad t \leq \frac{L_P}{2}$$

$$V_L = V_P\left(\frac{3}{L_P}t - \frac{2}{L_P^2}t^2\right) \qquad \text{for} \quad t \leq \frac{L_P}{2}$$

The time-value relationship imposes a financial penalty on projects that exceed potential launch dates with the value decreasing as the square of the time delay. The relationship shown in Figure 5.7 illustrates a loss of 50% of the value at a delay consuming approximately 20% of the potential life.

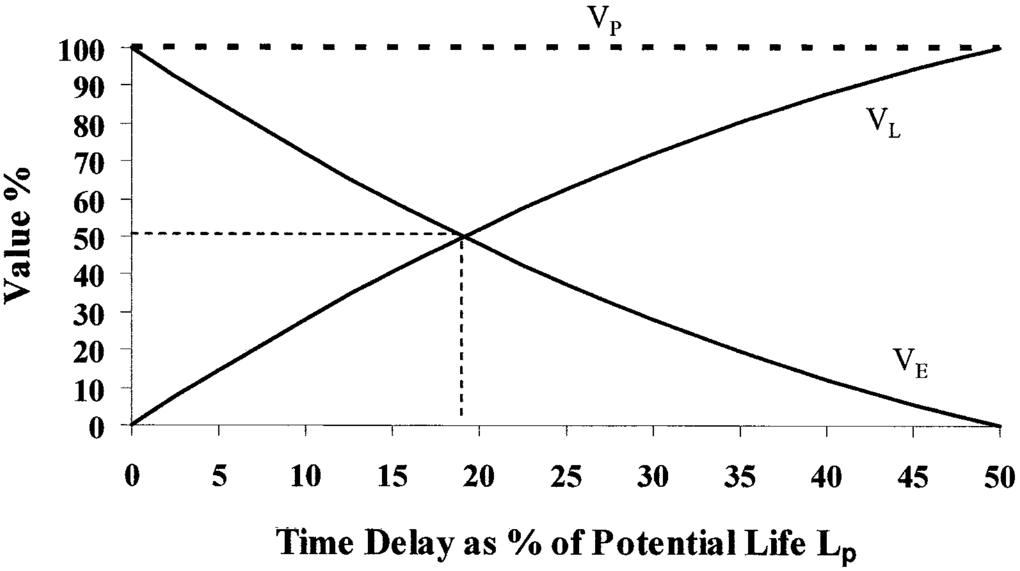


Figure 5.7 Time-Value model.

5.4.3 Net Present Value

The NPV of revenue calculations are based on the geometry of the triangular life cycle function and using the uniform gradient method. The method is applied as a uniformly increasing gradient (slope) on the left G_L and a uniformly decreasing gradient (slope) on the right G_R as shown in Figure 5.8. The gradient method is used to compute the NPV of the left gradient NPV_{LG} and right gradient NPV_{RG}. The method can be used to compute both the potential net present value NPV_P and the expected net present value NPV_E . The lost net present value NPV_L is simply the difference between NPV_P and NPV_E.

$$NPV_{LG}= P_{GL} (P/F, i, T)$$

$$\text{where } P_{GL}= G_L (P/G, i, n_L)$$

$$NPV_{RG} = P_{GRAD\text{-}RIGHT} (P/F, i, T+n_L)$$

$$\text{where } P_{GRAD\text{-}RIGHT} = R_P(P/A, I, n_R) +G_R (P/G, i, n_R)$$

The total net present value NPV_T is simply the sum of NPV_{LG} and $NPV_{RG.}$

$$NPV_T = NPV_{LEFT\text{-}GRAD} + NPV_{RIGHT\text{-}GRAD}$$

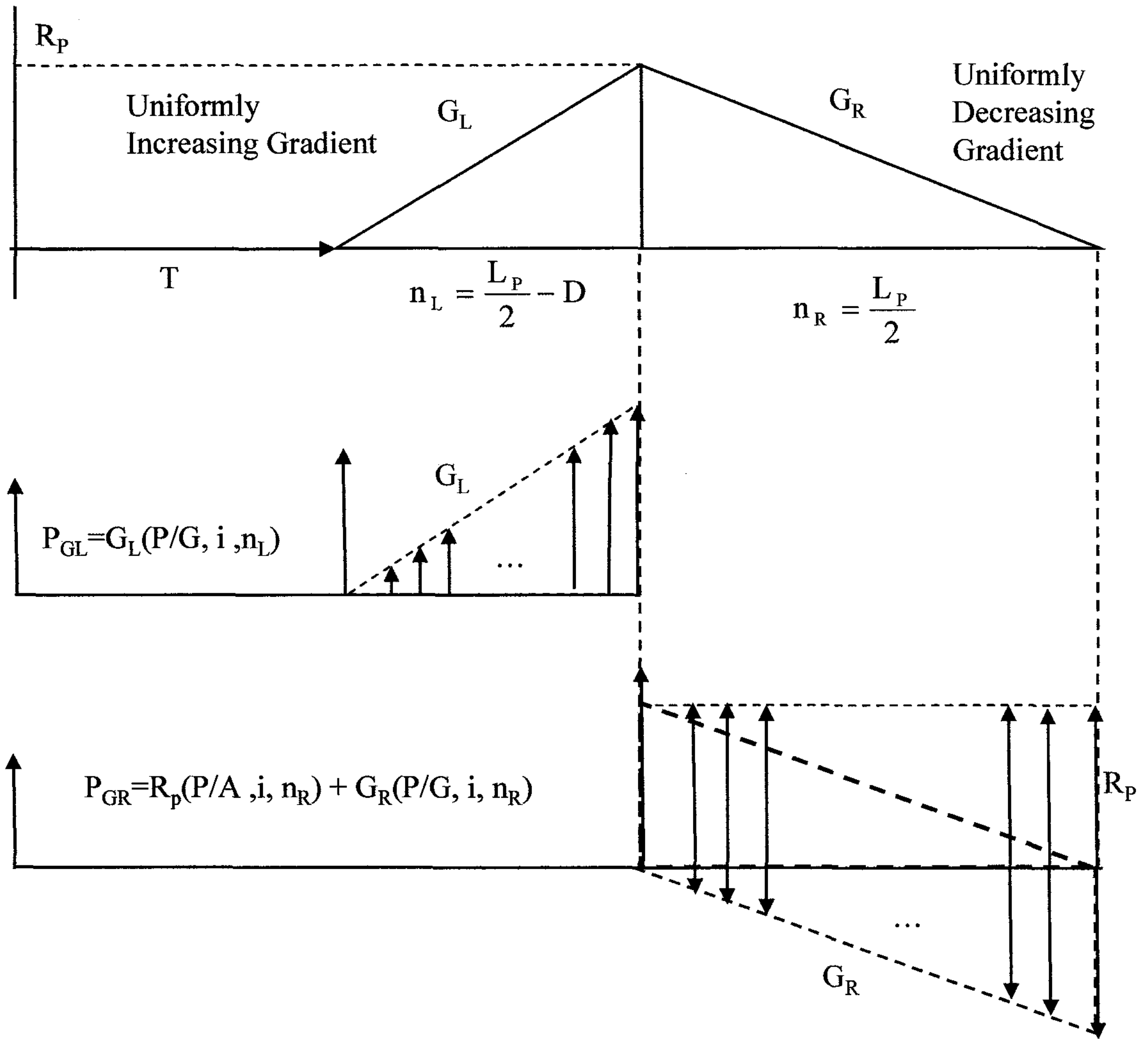

R_P
Uniformly Increasing Gradient
G_L
G_R
Uniformly Decreasing Gradient
T
$n_L = \frac{L_P}{2} - D$
$n_R = \frac{L_P}{2}$
G_L
$P_{GL}=G_L(P/G, i, n_L)$
...
$P_{GR}=R_p(P/A, i, n_R) + G_R(P/G, i, n_R)$
R_P
...
G_R

Figure 5.8 NPV computed using the Uniform Gradient Method.

5.5 Project Prioritization Method (PPM)

The NPD development organization is chartered with making tactical decisions regarding resource assignment in order to achieve market driven launch dates. The tactical decision facing the organization is to staff projects with resources in order to achieve the maximum time-value linkage for the portfolio. NPD organizations must carefully consider the complexity of the tasks and schedules and resource assignments across multiple projects contending for the same resources. Almost inevitably the NPD projects are delayed due to insufficient resources and scheduling constraints. The NPD organization must consider the impact delays to projects as well as the opportunity cost the portfolio value as a whole. Given the disparity between the strategic goals and what is practically achievable the NPD organization must use some criteria to decide how to assign resources most efficiently. It is not the intention of the NPD development organization to change the strategic priorities or to alter the portfolio selection. The development organization attempts to adhere closely to the strategic plan priorities even if achieving the priorities. However given the dynamic environment the resource assignment decisions will benefit from updated priorities based on robust scheduling estimates and an updated view of the time-value linkage model.

5.5.1 Project Value Metric

A Project Value Metric (PVM) is introduced to provide a measure for prioritizing projects in the NPD environment. The metric is a variation of the productivity index used to select the portfolio at the strategic level but is modified to account for the more accurate task and schedule estimates. The metric includes the time-value model and allows the practitioner to weight the model based on a more thorough perspective of the entire development program. A Project Value Metric (PVM) is first derived based on the Net Present Value of expected value and value lost using T_E the expected critical path duration as the common basis.

$$PVM = \frac{NPV_E + \alpha NPV_L}{T_E}(1 + \beta\gamma) \qquad 0 \le \alpha \le 1 \qquad 0 \le \beta \le 1$$

5.5.2 Prioritization Weighting Factors

The project planning environment may be subject to additional variation that would lead to a faster development time and recovery of the value lost. The recovery parameter α is selected to reflect the likelihood that the value lost (V_L) will be recovered. A weight of α = 0.20 would indicate that it is likely that 20% of the value lost will be recovered by schedule improvement and therefore 20% x V_L would be included in the PVM metric calculation. The parameter α allows the practitioner to include an optimistic adjustment to the overall program to reflect additional knowledge regarding the ability to improve schedules. For instance the organization may be in the process of securing additional funding or recruiting additional resources that will improve schedules so a higher α will be selected Additional information regarding the increased technical complexity or difficulty of the project set may result in a lower α. Other constraints such as availability of appropriately skilled technical staff may also influence α positively or negatively.

The factor (1+βγ) is the rate of further value loss. The γ loss rate factor is the normalized to the max loss rate across projects.

$$\gamma_i = \frac{V_L^{'}}{\max V_L^{'} \in i}$$

Where the loss rate is the derivative (18) of V_L the loss function (7) and evaluated at the current schedule view corresponding to t=D.

$$V'_L = \frac{V_P}{L_P}\left(3 - 4\frac{t}{L_P}\right)_{t=D}$$

The loss parameter β is multiplied by the normalized loss rate to reflect the likelihood of additional value lost due to incremental delay. For example a value of β = 0.10 indicates a 10% likelihood additional delay which will result across the portfolio. As a result the project value metric for an individual project is increased by 1.1 times the factor γ. The parameter β provides the practitioner with the ability to adjust the model based on pessimism that may result in further loss to the entire portfolio. Factors such as anticipated force reductions, diversion of resources and technical complexity within the portfolio may cause further delay and can be factored in by increasing the β factor. The factor β is constant across the portfolio however because γ is unique to each project. The weight of β can significantly alter the value of the metric.

5.5.3 Project Productivity Index (PPI)

The normalized Project Productivity Index (PPI) index is computed for each project as the ratio of PVM_i and the maximum PVM_{iMAX} value across all projects.

$$PPI_i = \frac{PVM_i}{PVM_{MAX \in i}}$$

The prioritized set of PPI ordered projects is give by (20):

$$\Pi_{\pi} = \{p_1, p_2, \cdots p_N\}$$

5.6 Multi-Project Prioritization Simulation

The NPD project prioritization model consists of straight-forward computations. All of the equations in this Chapter were programmed in Visual Basic for Applications in Excel. The program allows testing of the entire methodology including the Robust Scheduling Method, the Project Valuation Method and the Project Prioritization Method.

5.6.1 Program Example

A simple illustration is provided for six projects in Table 5.1. The projects were selected for similarity in size, scope and duration to allow the prioritization model to discern priorities across the non-dominated set of projects.

Table 5.1 shows the input for the project and the results of the Robust Scheduling Method. The input includes potential value V_p, time to potential launch T_p and potential life span L_p are provided as given input for each project. The Robust Scheduling Method utilizes the Critical Path Method (CPM) and the Critical Chain Buffer Management (CCBM) to derive values for Critical Path (CP) duration, Standard Deviation (SD), Bias Buffer (BB) and expected project duration (T_E).

Table 5.1 Prioritization Example Robust Scheduling Method (RSM).

Project Input				Robust Scheduling Method (RSM)			
Project	**VP**	**TP**	**LP**	**CP**	**SD**	**BB**	**TE**
	$	days	days	days	days	days	days
1	20,000,000	170	400	203	31	20	254
2	16,000,000	150	800	154	22	15	192
3	18,000,000	90	1200	147	29	15	191
4	16,000,000	110	400	133	43	13	189
5	20,000,000	180	800	224	36	22	283
6	18,000,000	130	1200	161	28	16	205

BB computation with ρ equal to 0.1

Table 5.2 Prioritization Example Project Valuation Method (PVM).

Project Valuation Method (PVM)						
Project	**D**	**VE**	**VL**	**NPVP**	**NPVE**	**NPVL**
	days	$	$	$	$	$
1	84	9,164,000	10,836,000	19,010,213	8,676,940	10,333,274
2	42	13,568,200	2,431,800	14,840,569	12,560,558	2,280,011
3	101	13,710,025	4,289,975	16,383,645	12,420,368	3,963,277
4	79	7,768,200	8,231,800	15,333,676	7,417,727	7,915,949
5	103	12,938,063	7,061,938	18,474,637	11,894,525	6,580,113
6	75	14,765,625	3,234,375	16,294,123	13,319,677	2,974,446

NPV computations with i =0.05.

Table 5.3 Prioritization Example Project Prioritization Method (PPM).

Project Prioritization Method (PPM)					
Project	**dVL/dt**	**gamma**	**PVM**	**PPI**	**Rank**
	$ per day		$ per day		
1	108,000	1	34164.496	0.5222824	6
2	55,800	0.5166667	65413.833	1	1
3	39,950	0.3699074	65006.258	0.9937693	2
4	88,400	0.8185185	39250.997	0.6000412	5
5	62,125	0.5752315	42027.848	0.6424918	4
6	41,250	0.3819444	64950.768	0.992921	3

PPI metric with α and β equal to 0.

Results for the Project Valuation Method (PVM) are shown in Table 5.2. The PVM starts by computing the Delay D which is subsequently used to calculate V_E and the value lost V_L using. The NPV values are computed using the uniform gradient method. Results for the Project Prioritization Method (PPM) are shown in Table 5.3. The PVM is used to compute the normalized rate factor gamma, the Project Prioritization Metric (PPM) and the Project Prioritization Index (PPI). The project rank is shown in last column of Table 5.3. The rank ordering of projects by arranging project from highest to lowest value of the PPI metric produces the set $\Pi = \{ p_2, p_3, p_6, p_5, p_4, p_1\}$.

A comparison of the PPI ranking to other potential prioritization criteria points to considerable differences. For example Project 2 originally had the lowest revenue potential V_P and the lowest NPV_P potential of any project yet after running the time-value priority model project 2 is ranked first ahead of all projects. The change for project 2 is attributable to the shorter project duration estimate and shorter delay resulting in preservation of greater expected revenue V_E. In addition project 1 had the highest potential NPV_P of all projects input to the model but after running the time-value model project 1 is ranked last after all projects because of the high value lost due to schedule delay. Thus the time value model effectively accounts for time-to-market delay and appropriately advances the rank of project based on productivity of scheduling per the PPI index.

5.6.2 Sensitivity to Prioritization Parameters

The PPM provides the project management practitioner with the flexibility to select the α and β parameters based on most current view of likelihood that the schedules can be improved or that further delays are anticipated across the portfolio. The guidelines for selecting α and β presented in this paper should be accompanied with an understanding of the sensitivity of these parameters. Figure 5.9 illustrates the sensitivity of the project ranking for the six projects in the prior simulation when values of α and β are allowed vary between 0 and 1.

1.0	{ 2, 6, 3, 4, 1, 5 }	{ 2, 3, 6, 4, 1, 5 }	{ 4, 1, 2, 3, 6, 5 }	{ 1, 4, 2, 3, 6, 5 }	{ 1, 4, 3, 2, 6, 5 }
0.8	{ 2, 6, 3, 4, 1, 5 }	{ 2, 3, 6, 4, 1, 5 }	{ 2, 4, 1, 3, 6, 5 }	{ 4, 1, 2, 3, 6, 5 }	{ 1, 4, 3, 2, 6, 5 }
0.5	{ 2, 6, 3, 4, 5, 1 }	{ 2, 3, 6, 4, 1, 5 }	{ 2, 3, 6, 4, 1, 5 }	{ 4, 1, 3, 2, 6, 5 }	{ 4, 1, 3, 2, 6, 5 }
0.2	C { 2, 6, 3, 5, 4, 1 }	D { 2, 3, 6, 4, 5, 1 }	{ 3, 2, 6, 4, 1, 5 }	{ 3, 4, 2, 6, 1, 5 }	{ 4, 3, 1, 6, 2, 5 }
0.0	A { 2, 3, 6, 5, 4, 1 }	B { 3, 6, 2, 4, 5, 1 }	{ 3, 6, 2, 4, 1, 5 }	{ 3, 6, 2, 4, 1, 5 }	{ 3, 4, 6, 2, 1, 5 }
β/α	0.0	0.2	0.5	0.8	1.0

Figure 5.9 Weighting factor α and β sensitivity.

This example illustrates 16 priority changes that occur as a result of changing the selection of α and β. The area highlighted in the lower left corner of the sensitivity matrix contains prioritized sets A, B, C and D and is worth noting in more detail. Region A is the baseline case where both α and β are zero such that the portfolio is not likely to be accelerated or to be delayed further. Region A essentially reflects a program at the expected value V_E and expected launch schedule T_E.

As α increases along the horizontal axis (and where β is zero) we observe changes in priority ranking from region A to B. This change reflects a greater likelihood that the overall program can be improved and a greater weight is placed on recovering lost value V_L. In this scenario where α is increasing we observe projects 3 and 6 advance in rank over project No. 2 and that project No. 5 advances in rank over priority No. 4. In both cases the projects having higher loss values ascend to higher rank because α is higher thus placing greater optimism that lost value can be recovered.

As β increases along the vertical axis (and where α is zero), there is a greater likelihood that program schedules will fall further behind greater emphasis is placed on pessimism regarding the rate of future loss. The increase of the loss factor β results in switching in priority ranking from region A to C where project No. 6 ascends in priority over project No. 3. We note that the rate of loss for project No. 6 ($\gamma = 0.382$) is slightly higher than the rate of loss of project No. 3 ($\gamma = 0.370$) therefore as β increases a higher weight is placed on project 6 as a scheduling priority in an effort to counter further loss.

As values of α and β both increase for example in region D, we observe additional changes in priority where project No. 4 advances in priority over project No. 5. Note that project No. 4 has a higher loss value (V_L = \$8.2M) than project No. 5 (V_L=\$7.0M) and project No. 4 has a higher rate of loss ($\gamma = 0.819$) than the rate of loss of project No. 5 ($\gamma = 0.575$). This is a combination effect where higher α and higher β place greater weight on recovering lost value in project No. 4 and at the same time countering further loss in project No. 4.

In the extreme regions Figure 5.9 further increases in α and β result in additional rank changes due to the same mechanism of placing more weight on loss recovery or on countering further loss. Extreme regions of the analysis with very high α and β may imply high uncertainty and potential instability in the development environment. Project managers should proceed with caution when considering both a high α and high β. A project manager would more typically select lower values of α and β (e.g. $\alpha+\beta \leq 1$).

CHAPTER 6

RESOURCE ASSIGNMENT METHOD

Chapter 5 and Chapter 6 satisfy Research Objective No. 3 by developing a decision model for SLTI NPD resource assignment in a multi-project environment. Chapter 5 developed a methodology for multi-project prioritization based on productivity metrics. Chapter 6 provides a methodology for resource across multiple NPD projects. The key input for Chapter 6 is the NPD project formulation described in Chapter 4 and the NPD Project Prioritization Model described in Chapter 5. The overall solution provides a multi-step approach as depicted in Figure 6.1 including project prioritization and resource assignment.

6.1 Resource Assignment Approach

The resource assignment and scheduling problem is solved through a series of iterative search algorithms using rules based heuristics. Resource assignment and scheduling is modeled in two phases including; 1) a constructive approach that employs priority rules for projects, tasks and resources to derive a feasible solution and 2) a improvement heuristic that considers the improved productivity that may be achieved by interchanging resources of differing skill levels and corresponding work rates. A computational analysis is carried out using multi-project data sets that simulate NPD project planning environment. Results point to significant productivity and efficiency gains that can be achieved through effective planning, prioritization and deployment of technical resources. In a real world setting product managers and engineering managers are constantly managing at a list of projects and opportunities to correlate and chart a path for these

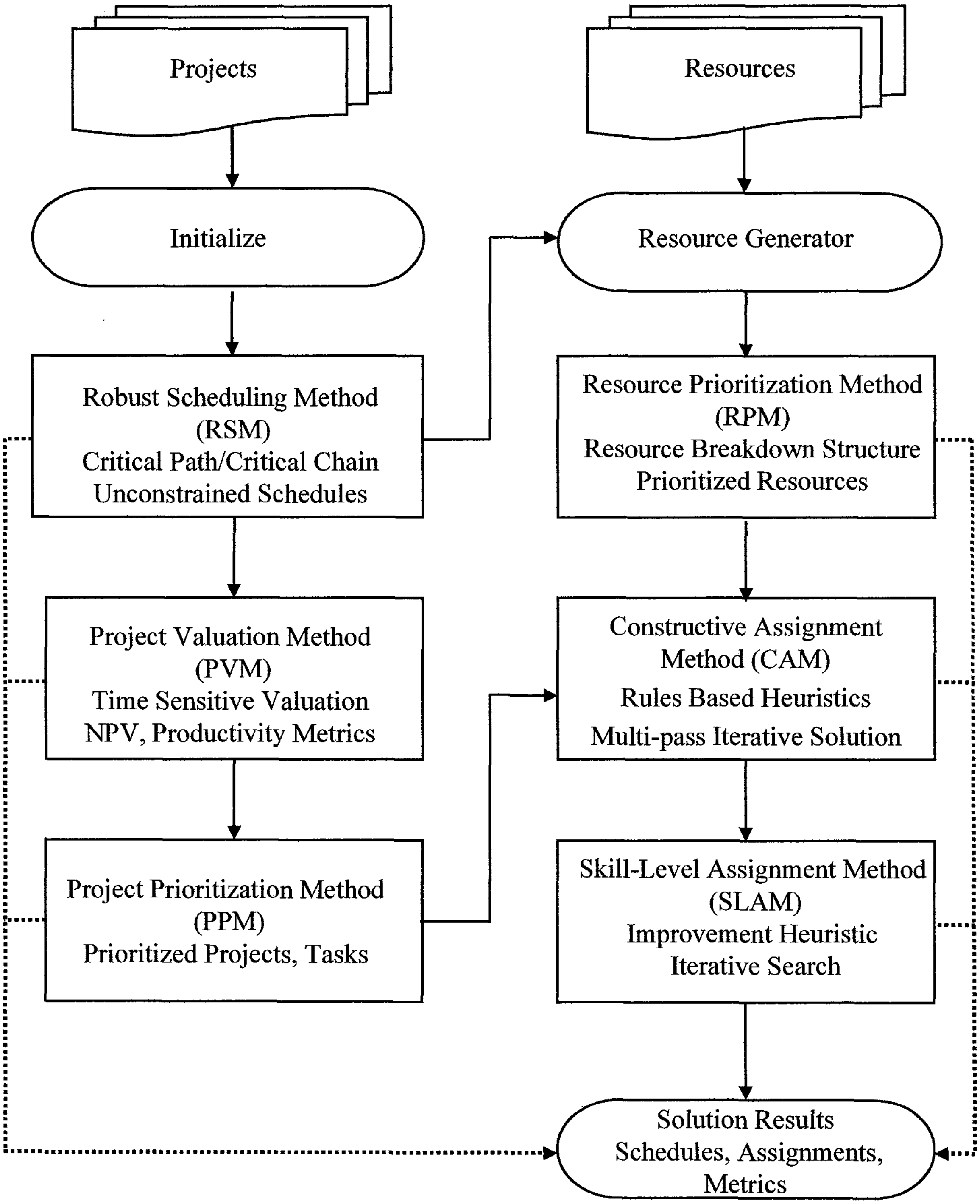

Projects
Resources
Initialize
Resource Generator
Robust Scheduling Method (RSM)
Critical Path/Critical Chain
Unconstrained Schedules
Resource Prioritization Method (RPM)
Resource Breakdown Structure
Prioritized Resources
Project Valuation Method (PVM)
Time Sensitive Valuation
NPV, Productivity Metrics
Constructive Assignment Method (CAM)
Rules Based Heuristics
Multi-pass Iterative Solution
Project Prioritization Method (PPM)
Prioritized Projects, Tasks
Skill-Level Assignment Method (SLAM)
Improvement Heuristic
Iterative Search
Solution Results
Schedules, Assignments,
Metrics

Figure 6.1 Overall solution method for Resource Assignment in SLTI NPD.

projects enabling introduction of new products. The tools available for managing multiple projects are often scattered including spreadsheets, project management software and text documents. In the case of multiple projects there are typically limited computational tools for comparing and estimating overlapping schedules. This research provides a model for evaluating projects and schedules in a multi-project environment.

6.2 Program Schedule Efficiency (PSE)

Solutions to the Resource Assignment problem are iterative in nature. As the solutions are generated a metric is needed to determine if an improvement has been made over previously generated solutions. A Program Schedule Efficiency (PSE) metric is introduced as a measure of efficiency improvement of one solution versus another. The PSE metric is based on the performance of achieving financial results relative to the maximum potential. The metric is a ratio of expected Net Present Value NPV_E of the current solution over the potential Net Present Value achieved NPV_P if the potential launch date T_p is achieved.

$$PSE = \frac{NPV_E}{NPV_P} \qquad 0 \le PSE \le 1$$

In theory it is possible to devise a schedule that will launch the product early ahead of the potential launch date Tp. Should that occur it might be possible to generate financial results early such that the expected NPV exceeds the potential. However T_p is defined in this research as the earliest date a product can be sold into the market. This definition imposes the restriction that expected NPV can be no greater than the potential NPV and the ratio will be at most one (1).

6.3 Resource Prioritization Method (RPM)

The resource inventory is used to initialize all of the resource assignment and scheduling methods described in this chapter. The resource inventory can be arranged randomly or alternatively it can be prioritized by skill level. A component of this investigation is to determine the impact of prioritization (vs. non-prioritization) of skill level on the overall performance of the schedule.

6.3.1 Resource Inventory

The resource assignment and scheduling methods outlined in this Chapter may be initialized with a manually developed resource data worksheet in Excel. The resource data file is constructed around the parameters described in Chapter 4 including:; M the Number of Resources, K the Number of Resource Types and L Number of Levels. For each of the M resources in a resource set the data file specifies the resource type g_m and the skill level h_m. Thus the resource inventory becomes a list with resource number, type and skill level.

To evaluated resource assignment methods at different levels of staffing the data file has been designed to allow entry of more than one resource set where each set is a different size. The resource sets shown in Figure 6.2 contains eight different resource sets each with the capability to complete the program of projects. The resource sets are intended to schedule a defined program consisting of three required resource types (K=3) therefore each resource set mush have at least one of each type in order to achieve a feasible solution. By definition resource sets M1 and M2 would consist of fewer than the minimum required three resource types and therefore are not feasible. The data file allows the creation of larger resource sets within the limited of VBA in Excel. The data file can be manually initialized to contain any distribution of resource types or skill levels. Furthermore the file structure can easily be scaled to incorporate additional resource types and skill levels of additional defined programs.

Resource	M=1		M=2		M=3		M=4		M=5		M=6		M=7		M=8		M=9		M=10	
m	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM
1					1	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1
2					2	2	2	1	2	1	1	2	1	2	1	2	1	2	1	2
3					3	2	2	2	2	2	2	1	2	1	2	1	1	3	1	3
4							3	2	2	3	2	2	2	2	2	2	2	1	2	1
5									3	2	2	3	2	3	2	2	2	2	2	2
6											3	2	3	1	2	3	2	3	2	2
7													3	2	3	1	2	3	2	3
8															3	2	3	2	3	1
9																	3	3	3	2
10																			3	3

Figure 6.2 Sample resource data file generated manually.

6.3.2 Resource Generator

The resource inventory developed manually in the previous section allows specific distributions of types and levels to be created and tested against resource assignment methods. Alternatively for large scale computational analysis a resource generator is introduced that automatically generates the inventory for assignment. The generator uses information from the Robust Scheduling Method (RSM) to size the requirements for each type and then to generates a resource type distribution that matches the demand requirements from the actual tasks in the projects. In addition the resource generator produces a uniform distribution of skill levels across each of the resource sets.

The resource generator initializes a resource set with the minimum number of resources required to achieve a feasible solution which equals K the number of required resources. The generator then analyzes the demand for resources of each type and generates successive resource sets to select an additional resource of greatest need to balance the distribution. The initial set consists of K resources all having resource level two (2).As a resource is selected to initialize an incrementally larger set the skill level is selected in a progression to establish a uniform distribution of skill levels. The resource generator can essentially automatically generate a consistent resource inventory at any size for computational purposes. An example of a resource inventory produced with the resource generator is shown in Figure 6.3.

Resource	M=1		M=2		M=3		M=4		M=5		M=6		M=7		M=8		M=9		M=10	
m	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM
1					1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
2					2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3					3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
4							2	1	2	1	2	1	2	1	2	1	2	1	2	1
5									2	2	2	2	2	2	2	2	2	2	2	2
6											1	3	1	3	1	3	1	3	1	3
7													3	1	3	1	3	1	3	1
8															2	2	2	2	2	2
9																	2	3	2	3
10																			1	1

Figure 6.3 Sample resource data sets.

6.3.3 Resource Prioritization

The resource inventory can be arranged randomly or alternatively it can be prioritized by skill level. A component of this investigation is to determine the impact of prioritization (vs. non-prioritization) of skill level on the overall performance of the schedule. The Resource Prioritization Method (RSM) simply arranges resources by type and by skill level. Resources generated in a manual data file or by the resource generator may be sorted in ascending order. An example of resource sets that have been sorted by type and by level is shown in Figure 6.4.

Resource m	M=1		M=2		M=3		M=4		M=5		M=6		M=7		M=8		M=9		M=10	
	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM
1					1	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1
2					2	2	2	1	2	1	1	2	1	2	1	2	1	2	1	2
3					3	2	2	2	2	2	2	1	2	1	2	1	1	3	1	3
4							3	2	2	3	2	2	2	2	2	2	2	1	2	1
5									3	2	2	3	2	3	2	2	2	2	2	2
6											3	2	3	1	2	3	2	3	2	2
7													3	2	3	1	2	3	2	3
8															3	2	3	2	3	1
9																	3	3	3	2
10																			3	3

Figure 6.4 Sample of the Resource Prioritization Method (RSM).

6.4 Constructive Scheduling Heuristics

Resource assignment and scheduling is modeled in two phases including; 1) a Constructive Assignment Method (CAM) that employs priority rules for projects, tasks and resources to develop a feasible solution schedule and 2) a improvement heuristic that considers the improved productivity that may be achieved by interchanging resources of differing skill levels and corresponding work rates.

The objective of the CAM method is to build a feasible solution to the Resource Assignment problem. The feasible solution must satisfy time and resource constraints and develop complete schedules for each project. The CAM solution becomes the basis for investigating further improvements. As part of this research three different constructive solutions have been modeled including; 1) Sequential Scheduling Method, 2) Critical Scheduling Method and 3) Early Scheduling Method.

6.4.1 Sequential Scheduling Method (SSM)

The Sequential Scheduling Method (SSM) assigns resources in order of project priority and within project the sequence of task numbering. Figure 6.5 illustrates a simple example for three projects each consisting of five tasks.

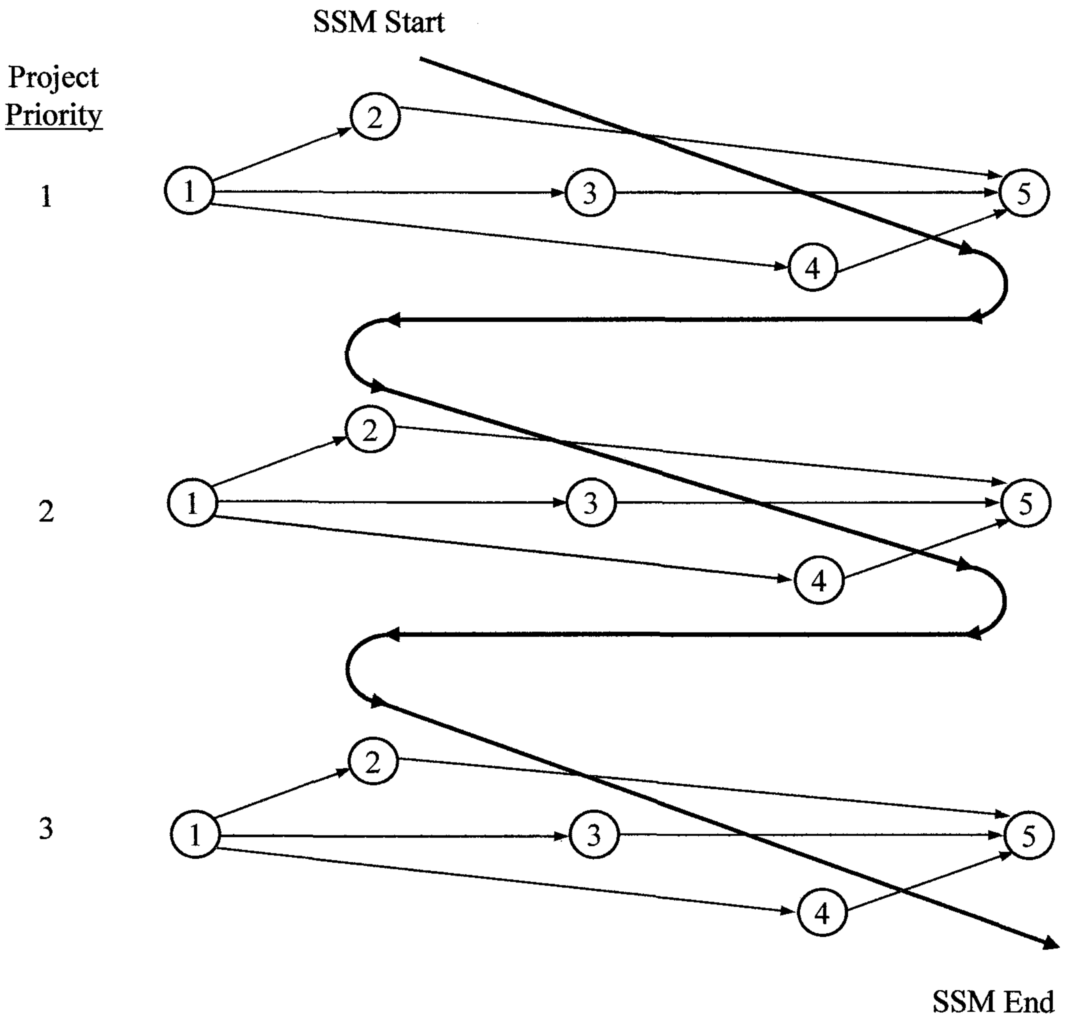


Figure 6.5 Sequential Scheduling Method (SSM).

The Sequential Scheduling Method initializes projects in priority order and assigns resources to each project one at a time. The rule Sequential Task First (STF) is used within each project to assign resources to tasks one at a time. In general tasks have been numbered from left to right with the requirement that for any task lower task numbers are predecessors and higher task numbers are successors. No priority beyond the sequential task number is implied by the STF rule. The method consists of two primary passes; 1) first pass to construct a Time Constrained Solution (TCS) and 2) second pass to construct a Resource Constrained Solution (RCS) where all tasks are assigned including tasks requiring delay.

The first pass to construct the TCS follows the sequence exactly as described. All tasks are initialized with a Start (ST) date according to ES dates that result from the Robust Scheduling Method. The algorithm attempts to assign resource m from the inventory by first checking that the resource type meets the required type (i.e. $g_m = f_{ij}$) and that the resource is available in the period ($A_{jt} \neq m$). If the resource does not match type or is unavailable then the algorithm attempts to assign the next resource in the inventory. The resources can be assigned in random order or in order of skill level priority. If there is no resource that can satisfy assignment constraints directly then if slack is available it is used. Slack at a particular task is incremented one day and the SSM attempts to assign each resource in turn. The method continues to use slack in the network while attempting to assign resources to tasks without changing the end date of the final task. If slack is used the algorithm updates Start (ST) and Finish (FN) dates accordingly in the task and subsequent tasks. At completion of the first pass a Time Constrained Solution has been formulated that produces a schedule for projects without changing the end finish date for any project. Projects that may be completely assigned have Start (ST) and Finish (FN) dates assigned for every task. If a project is completely assigned then the ST and FN as well as assignments on that project accommodate all constraints in the Robust Scheduling Method and therefore become part of the feasible solution. At the completion of the first pass some projects may be only partially assigned depending on the quantity of resources available. If a project is not completely assigned then that project is re-initialized by removing any partial assignments and setting an indicator to reassign the entire project in the second pass.

The second pass introduces a time lag and delay as required to force assignments to tasks and projects. Only those projects not assigned completed by the Time Constrained Solution in the first pass participate in the second pass. After re-initializing partially assigned projects the second pass follows the same priority sequence as the first pass. The second pass any task that remains unassigned after consuming any available slack is assigned a lag. The lag is incremented one day at a time and at each increment the algorithm attempts to assign a resource from the available inventory. When a time period is reached where a resource can be assigned the lag is integrated into the schedule by updating start (ST) and finish (FN) dates for the task any subsequent tasks in the network. Thus any task requiring a lag essentially builds in a delay for that tasks, successor tasks and for the entire project. Upon completion of the second pass all tasks and projects have been assigned resources, start (ST) dates and finish (FN) dates. The Resource Constrained Solution (RCS) now becomes the SSM feasible solution although some project may no longer satisfy time constraints. Note that the RCS solution contains the TCS as a subset. In addition should sufficient resources be available the RCS and TCS are identical.

At completion of the SSM the Project Valuation Method is updated to compute net present value and the Program Schedule Efficiency (PSE). Since the SSM is usually the first feasible solution in the overall method the RCS is automatically used to initialize the Best Constructive Solution (BCS).

6.4.2 Critical Scheduling Method (CSM)

The Critical Scheduling Method (SSM) assigns resources to critical tasks on projects first. Figure 6.6 illustrates a simple example of the CSM for three projects each consisting of five tasks.

The Critical Scheduling Method initializes projects in priority order and uses the Critical Task First (CTF) rule to assign resources to critical tasks across all projects before scheduling non-critical tasks on any project. The CTF order of assignment priority is; 1) Critical Tasks (CT), 2) Sub-Critical (SC) tasks and 3) non-critical tasks. The algorithm is completed in three passes; 1) a first pass to formulate a Time Constrained Solution, 2) a second pass to construct a Critically Constrained Solution (CCS) and 3) a third and final pass to construct a Resource Constrained Solution (RCS).

The first pass is used to assign critical tasks in order of project priority to construct a TCS. The algorithm identifies critical tasks on each project and assigns critical tasks within projects in sequential order. All critical tasks are initialized with a Start (ST) date according to ES dates that result from the Robust Scheduling Method. The algorithm attempts to assign resource m from the inventory by first checking that the resource type meets the required type (i.e. $g_m = f_{ij}$) and that the resource is available in the period ($A_{jt} \neq m$). If the resource does not match type or is unavailable then the algorithm

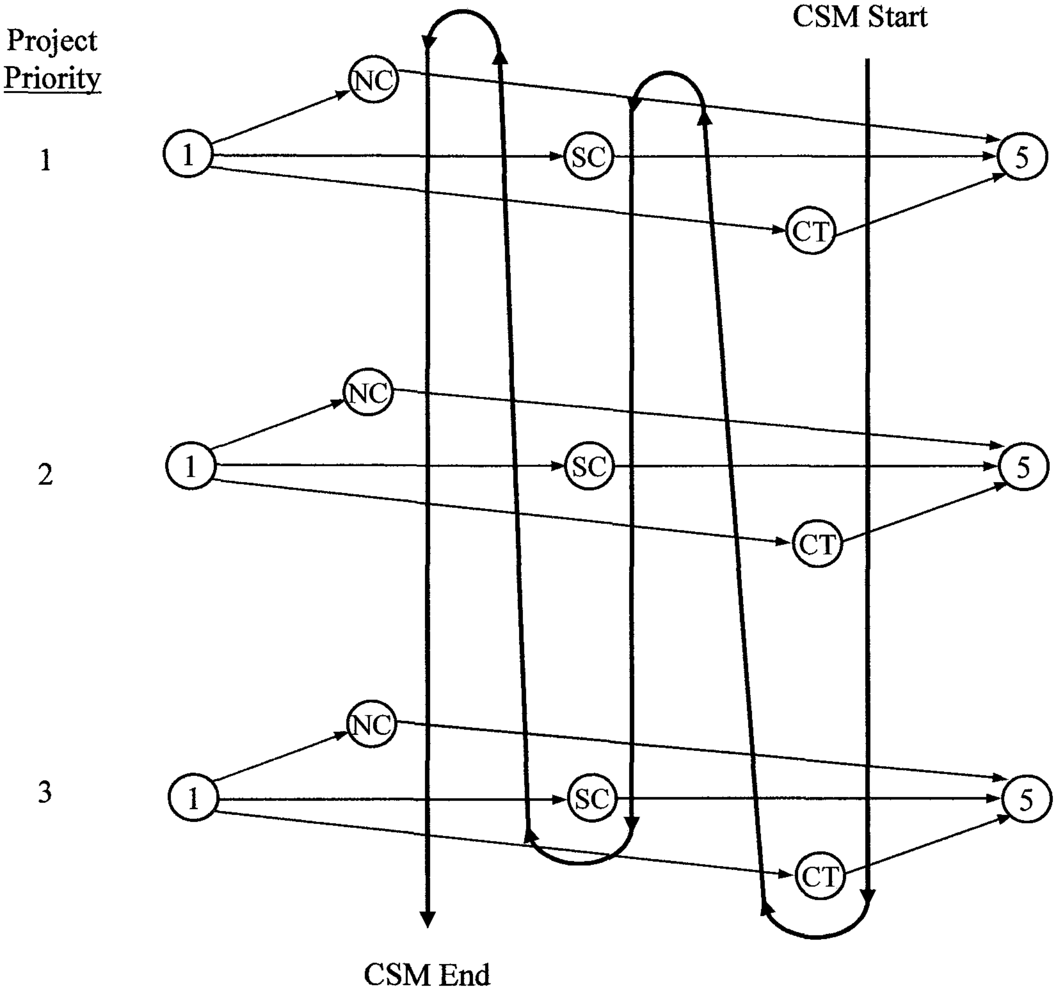


Figure 6.6 Critical Scheduling Method (CSM)).

attempts to assign the next resource in the inventory. The resources can be assigned in random order or in order of skill level priority. If there is no resource that can satisfy assignment constraints then it moves to the next critical task. Slack is not available on critical tasks so it is not considered. After attempting to assign all critical tasks the algorithm next attempts to assign resources to sub-critical tasks. Sub-critical tasks are defined as those non-critical tasks that have a high potential to become critical when changes occur in the network. Sub-critical tasks are tasks having limited slack and are identified as follows:

$$SC_{ij} = \begin{cases} 0 \text{ if } TS_j > \delta \; CP_i \\ 1 \text{ if } TS_j \leq \delta \, CP_i \end{cases} \tag{6.2}$$

$$0 \leq \delta \leq 1$$

Sub-critical tasks are assigned in order of project priority and in order of sequential task number within a particular project. Sub-critical tasks while having limited slack are treated as though no slack is available. Thus no slack is used in attempting to assign a resource to a critical task on the first pass. Finally the algorithm assigns resources to non-critical tasks which are all remaining tasks. Assignment of non-critical tasks may include slack if it is available. At completion of the first pass a Time Constrained Solution has been formulated that produces a schedule for projects without changing the end finish date for any project. Projects that may be completely assigned have Start (ST) and Finish (FN) dates assigned for every task. If a project is completely assigned then the ST and FN as well as assignments on that project accommodate all constraints in the Robust Scheduling Method and therefore become part of the feasible solution. At the completion of the first pass some projects may be only partially assigned depending on the quantity of resources available. If a project is not completely assigned then that project is re-initialized for reassignment in a subsequent pass.

The CSM algorithm incorporates a second pass to iteratively reassign resources from partially assigned lower priority projects to critical, sub-critical and non-critical tasks on partially assigned higher priority projects. This approach formulates the Critically Constrained Solution (CCS) with the available resources. Once resources are cleared from partially assigned lower priority projects the algorithm essentially repeats the first pass in project priority order starting with the highest priority partially assigned project. The second pass essentially extends the Critical Task Rule to a Critical Project Rule that fully satisfies all assignments of the most critical projects first.

After the first and second pass a third pass is completed to assign resources to partially assigned projects. All partially assigned projects are first re-initialized by removing partial assignments. The SSM is then applied to any project without assignments in priority order while adding lags and delay as necessary to the lower priority projects. The lag is incremented one day at a time and at each increment the algorithm attempts to assign a resource from the available inventory. When a time period is reached where a resource can be assigned the lag is integrated into the schedule by updating start (ST) and finish (FN) dates for the task any subsequent tasks in the network. Thus any task requiring a lag essentially builds in a third second pass all tasks and projects have been assigned resources, start (ST) dates and finish (FN) dates. The Resource Constrained Solution (RCS) now becomes the CSM feasible solution although it may no longer satisfy time constraints. Note that the RCS solution contains both the TCS and CCS as subsets. In addition should sufficient resources be available the RCS, TCS and CCS all will be identical.

At completion of the CSM the Project Valuation Method is updated to compute net present value and the Program Schedule Efficiency (PSE). The PSE is tested against prior generated feasible solution (using the SSM) and if the PSE is higher for CSM the solution is saved as the Best Constructive Solution (BCS).

6.4.3 Early Scheduling Method (ESM)

The Early Scheduling Method (SSM) uses the Early Task First (ETF) rule to pre-sort all tasks in all projects according to the Earliest Start (ES) date. Tasks are sorted from lowest to highest ES date. Figure 6.7 illustrates an example for three projects each consisting of five tasks.

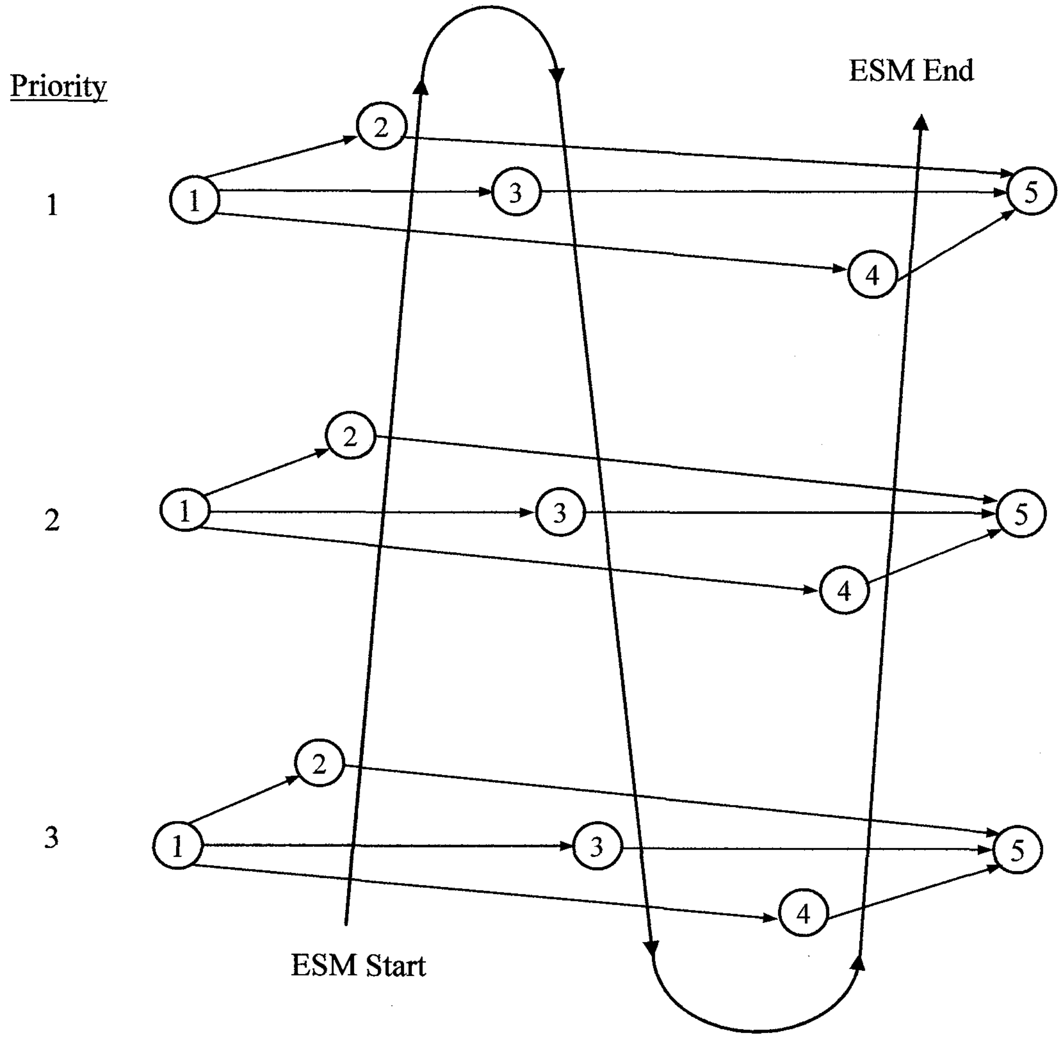


Figure 6.7 Early Scheduling Method (ESM)).

The method consists of two primary passes; 1) first pass to construct a Time Constrained Solution (TCS) and 2) second pass to construct a Resource Constrained Solution (RCS) where all tasks are assigned including tasks requiring delay.

The first pass to construct the TCS follows assigns tasks in ES priority order starting with the lowest duration task. Tasks are initialized with a Start (ST) date according to ES dates that result from the Robust Scheduling Method. The algorithm attempts to assign resource m from the inventory by first checking that the resource type meets the required type (i.e. $g_m = f_{ij}$) and that the resource is available in the period ($A_{jt} \neq$ m). If the resource does not match type or is unavailable then the algorithm attempts to assign the next resource in the inventory. The resources can be assigned in random order or in order of skill level priority. If there is no resource that can satisfy assignment constraints directly then if slack is available it is used. Slack at a particular task are incremented one day and the SSM attempts to assign each resource in turn. The method continues to use slack in the network while attempting to assign resources to tasks without changing the end date of the final task. If slack is used the algorithm updates Start (ST) and Finish (FN) dates accordingly in the task and subsequent tasks. At completion of the first pass a Time Constrained Solution has been formulated that produces a schedule for projects without changing the end finish date for any project. Projects that may be completely assigned have Start (ST) and Finish (FN) dates assigned for every task. If a project is completely assigned then the ST and FN as well as assignments on that project accommodate all constraints in the Robust Scheduling Method and therefore become part of the feasible solution. At the completion of the first pass some projects may be only partially assigned depending on the quantity of resources available. If a project is not completely assigned then that project is re-initialized by removing any partial assignments and setting an indicator to reassign the entire project in the second pass.

The second pass introduces a time lag and delay as required to force assignments to tasks and projects following the SSM. Only those projects not assigned completed by the Time Constrained Solution in the first pass participate in the second pass. After re-initializing partially assigned projects the second pass follows the same priority sequence as the first pass. The second pass any task that remains unassigned after consuming any available slack is assigned a lag. The lag is incremented one day at a time and at each increment the algorithm attempts to assign a resource from the available inventory. When a time period is reached where a resource can be assigned the lag is integrated into the schedule by updating start (ST) and finish (FN) dates for the task any subsequent tasks in the network. Thus any task requiring a lag essentially builds in a delay for that tasks, successor tasks and for the entire project. Upon completion of the second pass all tasks and projects have been assigned resources, start (ST) dates and finish (FN) dates. The Resource Constrained Solution (RCS) now becomes the feasible ESM solution although it may no longer satisfy time constraints. Note that the RCS solution contains the TCS as a subset. In addition should sufficient resources be available the RCS and TCS are identical.

At completion of the ESM the Project Valuation Method is updated to compute net present value and the Program Schedule Efficiency (PSE). The PSE is tested against prior generated feasible solution (using the SSM) and if the PSE is higher for ESM the solution is saved as the Best Constructive Solution (BCS).

6.5 Skill Level Improvement Heuristics

Resource assignment and scheduling is modeled in two phases including; 1) a constructive approach that employs priority rules for projects, tasks and resources to derive a feasible solution and 2) a improvement heuristic that considers the improved productivity that may be achieved by interchanging resources of differing skill levels and corresponding work rates. In section 6.2 a Best Constructive Solution (BCS) was developed using several heuristic assignment rules such as Sequential Task First, Critical Task First and Earliest Task First. Thus far the constructive approach has assumed that all resources are the same skill level and work at the same average work rate. The assumption that all resources are the same level and work at the same rate is relaxed in this section. An improved heuristic solution is developed that considers productivity gains that may be achieved associated with judicious assignment according to skill level and applying the corresponding work rates. A Level Constrained Solution (LCS) is developed by assigning each task the skill level of the resource already assigned in the Best Constructive Solution. Then the Skill Level Assignment Method (SLAM) is used to consider substitution of resources between tasks and projects in order to achieve productivity gains.

6.5.1 Level Constrained Solution

The BCS contains the solution with the best assignments and schedules as measured by highest PSE performance. The BCS solution assumes that all resources are the same skill level and work at an average work rate. The Level Constrained Assignment Method assigns each task a skill level corresponding to the resource already assigned to that task in the BCS. The work rates are then applied according to the work rate transformation in 4.3.3. An entirely new schedule is then developed associated with the task duration transformation. The new schedule is a Level Constrained Solution (LCS) with new project durations and a new level of performance characterized by PSE. It is important to note that the LCS could be higher or lower than the BCS depending on the distribution of tasks and placement of the various skill levels on the tasks.

6.5.2 Skill Level Assignment Method (SLAM)

Then Skill Level Assignment Method (SLAM) starts with the LCS and considers substitution of resources between tasks and projects in order to achieve productivity gains. As depicted in Figure 6.8, resources of the same type but differing skill levels are distributed among several projects. The opportunity arises to consider substituting a resource of one level already assigned to on task for a resource of differing level assigned to another task. The decision analysis is relative specifically where tasks overlap such that a resource contention is created. This research addresses the case of three skill levels ($L=3$, $q_1=1$, $q_2=2$, $q_3=3$) as case in point. However nothing restricts expansion of the methodology to assign or substitute additional levels. The methodology starts by identifying suitable pairs that should be considered for substitution. Suitable resource (task) pairings should; 1) have the same Resource Type or Resource Type requirement, 2) have different skill levels and 3) occur in overlapping time periods.

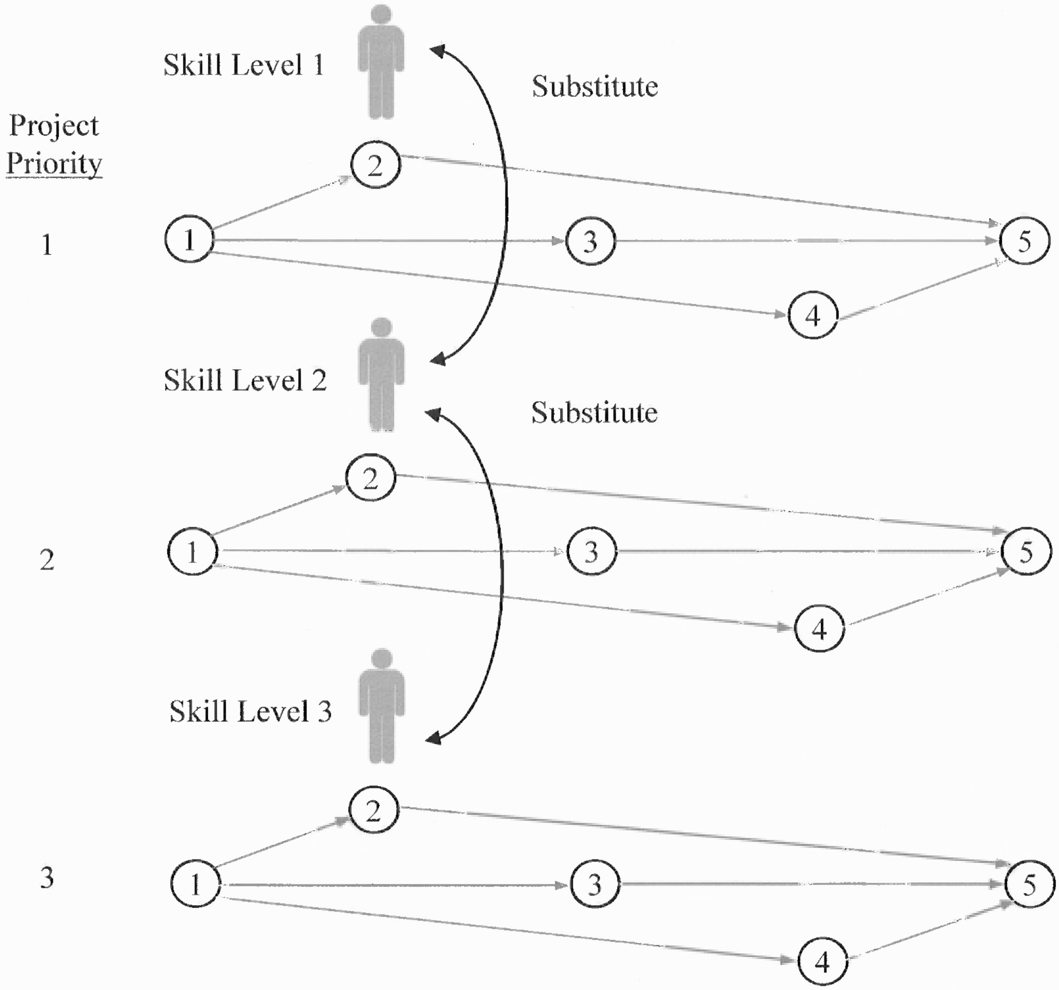


Figure 6.8 Skill Level Assignment Method (SLAM).

This research limits the number of skill levels to three in order to reduce the combinatorial complexity and the complexity is further reduced by considering only two pairings. The algorithm considers potential substitution of resource level one (1) for level (2) and substitution of resource level two (2) for level three (3). While the algorithm does not directly consider substitution of a resource level one (1) with a level (3), the program and computational analysis that follow executes the 1:2 and 2:3 substitutions sequentially thereby effectively including a 1:3 potential substitution in the methodology.

The SLAM methodology proceeds to select an appropriate pair for trial substitution. A trial exchange of assignment, skill level and durations is completed for the pair. All other resource assignments are then removed in the process of re-initializing for reassignment. It is important to note that skill level assignments are not reset. Skill level assignments on each task remain in tact effectively fixing the durations for all other tasks. Because the durations have transformed for the substitution pair a complete reschedule is required. A trial RSM using the transformed durations is developed and then SSM reassignment occurs for each projects. The algorithm develops an updated solution with pair substitution and new assignments and schedules. A PSE is computed for the new SLAM solution and if an improvement has been achieved the substitution is implemented in the updated solution and that SLAM solution becomes the new best solution. If no improvement is returned then the trial solution is dropped, original assignments reinstated and the existing LCS solution is carried forward as the best solution.

An exhaustive iterative search and trial substitution is completed for all 1:2 pairs that satisfy potential substitution criteria. The schedule and assignments are updated for each improvement in the PSE criteria. Once all qualified 1:2 substitutions are evaluated the algorithm completes an exhaustive iterative search to evaluate all 2:3 pairs that satisfy potential substitution criteria. Again schedules and assignments are updated should the search result improvement to the PSE criteria. Following the 2:3 pair search the best SLAM solution is obtained and becomes the recommended overall program solution along with complete resource assignments and project schedules.

CHAPTER 7

DECISION SUPPORT TOOL AND EXPERIMENTAL RESULTS

Chapter 7 describes the Resource Assignment and Multi-Project Scheduling (RAMPS) program developed to satisfy Research Objective No. 4 and provide a decision tool to assist with resource assignment in SLTI NPD. The RAMPS program automates the entire methodology presented in this dissertation and solves the NPD multi-project resource assignment problem using a combination of traditional project management and priority rule heuristics. The RAMPS program is design following the formulation described in Chapter 4 and embodies the algorithms described in Chapter 5 and Chapter 6. A description of RAMPS program is contained in this chapter along with results of the computational experiments designed to evaluate the program.

7.1 Resource Assignment and Multi-Project Scheduling Program

7.1.1 Program Organization

The RAMPS program was developed in Visual Basic for Applications (VBA) in Excel convenient access and universal acceptance for users interested in implementing the code. Input and output files can easily be formatted in worksheets in Excel. Flow charts are provided in Appendix A for each of the ten VBA procedures, including:

100 – Robust Scheduling Method
200 – Project Valuation Method
300 – Project Prioritization Method
400 – Resource Prioritization Method
500 – Sequential Scheduling Method
600 – Critical Scheduling Method
700 – Early Scheduling Method
800 – Level Scheduling Method
900 – Skill Level Assignment Method 1:2
1000 – Skill Level Assignment Method 2:3

The RAMPS program code listed Appendix C is organized into VBA procedures according to the line numbers referenced in Figure 7.1.

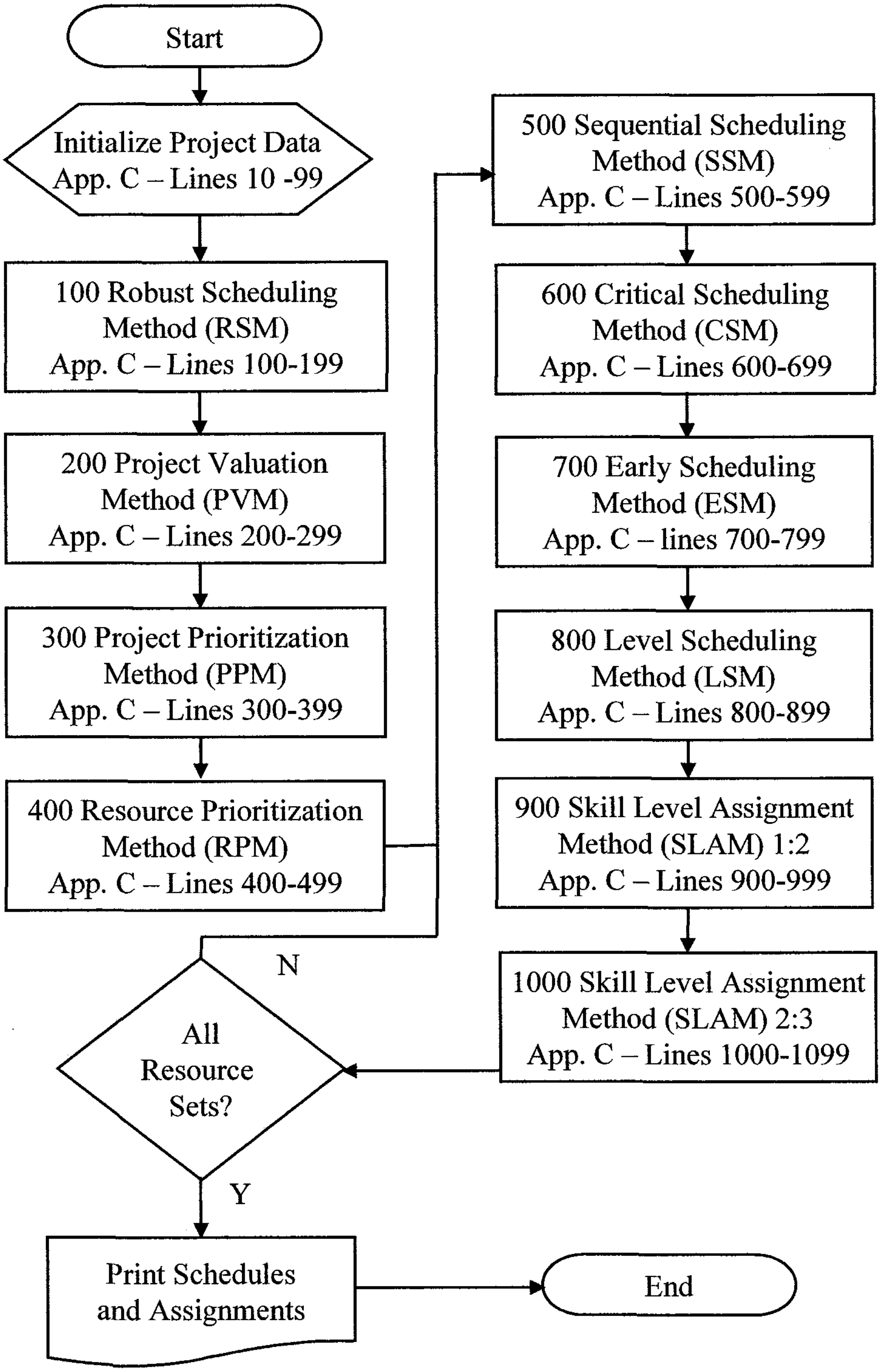

Start
Initialize Project Data
App. C – Lines 10 -99
100 Robust Scheduling Method (RSM)
App. C – Lines 100-199
200 Project Valuation Method (PVM)
App. C – Lines 200-299
300 Project Prioritization Method (PPM)
App. C – Lines 300-399
400 Resource Prioritization Method (RPM)
App. C – Lines 400-499
500 Sequential Scheduling Method (SSM)
App. C – Lines 500-599
600 Critical Scheduling Method (CSM)
App. C – Lines 600-699
700 Early Scheduling Method (ESM)
App. C – lines 700-799
800 Level Scheduling Method (LSM)
App. C – Lines 800-899
900 Skill Level Assignment Method (SLAM) 1:2
App. C – Lines 900-999
1000 Skill Level Assignment Method (SLAM) 2:3
App. C – Lines 1000-1099
All Resource Sets?
N
Y
Print Schedules and Assignments
End

Figure 7.1 Program structure.

7.1.2 Project Input Data

The program structure is initialized with project data per the formulation in Chapter 4. The project network example shown in Figure 7.2 consists of ten activities where activities one (1) and ten (10) are artificial and have zero duration. This example is carried forward to illustrate the input sequence. The corresponding input data file for the RAMPS program is shown in Figure 7.3

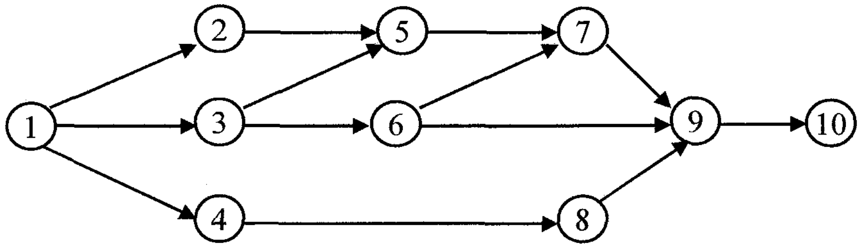


Figure 7.2 Project network example.

i	1		PROJECT 1 - INPUT							
VP	$17,000,000									
TP	90									
LP	400									
j	1	2	3	4	5	6	7	8	9	10
DM	0	14	28	21	14	28	21	35	21	0
DL	0	21	35	35	28	35	35	49	28	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	1	0	1	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Figure 7.3 Project input data example.

The input data file provides all the necessary data to initialize project data including project number (i), potential value (VP), potential time to launch (TP) and potential life (LP). The task data includes mean duration (DM), low risk duration (DL), required resource type (FJ) and initial lag (LG). Initial lags are shown as zero but could be initialized to reflect a specified start time. The precedence matrix (PRC) is also contained in the input data to provide a complete characterization of the predecessor and successor relationships for each task.

7.1.3 Resource Input Data

The RAMPS program can be executed with one set of resources or with multiple resource sets of difference sizes. Processing multiple resource sets allows the program to test how the NPD assignment and scheduling performance changes with resource sets of different sizes, different type distributions and different skill level distributions. As discussed in 6.2 the resource set(s) may be read from a manually generated input file or may be generated automatically using the resource generator feature. Each resource set has a unique characterization of resource types and skill levels. Resources can be applied to the RAMPS program sequentially or alternatively resources can be presorted and prioritized by type and skill level. The Resource Prioritization Method (RPM) implemented in procedure 400 can be used to prioritize resources by skill level for input to the RAMPS program. An example of prioritized resource data sets generated automatically for three (3) resource types ranging in size from three (3) to ten (10) are shown in Figure 7.4

400 - RESOURCE PRIORITIZATION METHOD (RPM)

Resource m	M=1		M=2		M=3		M=4		M=5		M=6		M=7		M=8		M=9		M=10	
	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM	GM	HM
1					1	2	1	1	1	2	1	1	1	1	1	1	1	1	1	1
2					2	2	2	1	2	1	1	2	1	2	1	2	1	2	1	2
3					3	2	2	2	2	2	2	1	2	1	2	1	1	3	1	3
4							3	2	2	3	2	2	2	2	2	2	2	1	2	1
5									3	2	2	3	2	3	2	2	2	2	2	2
6											3	2	3	1	2	3	2	3	2	2
7													3	2	3	1	2	3	2	3
8															3	2	3	2	3	1
9																	3	3	3	2
10																			3	3

Figure 7.4 Resource input data example.

7.1.4 Program Solution

The Robust Scheduling Method (RSM) implemented in procedure 100 incorporates the entire methodology described in 5.3 and computes baseline scheduling results used in project prioritization and throughout all subsequent scheduling procedures. An example of RSM results for one project is shown in Figure 7.5.

```
100 - ROBUST SCHEDULING METHOD (RSM)
CRITICAL PATH/CRITICAL CHAIN SCHEDULING

Project   1    1    1    1    1    1    1    1    1    1
Task      1    2    3    4    5    6    7    8    9   10
DM        0   14   28   21   14   28   21   35   21    0
DL        0   21   35   35   28   35   35   49   28    0
FJ        0    1    1    2    2    2    2    3    3    0
LG        0    0    0    0    0    0    0    0    0    0
ES        0    0    0    0   28   28   56   21   77   98
EF        0   14   28   21   42   56   77   56   98   98
LS        0   28    0   21   42   28   56   42   77   98
LF        0   42   28   42   56   56   77   77   98   98
TS        0   28    0   21   14    0    0   21    0    0
FS        0   14    0    0   14    0    0   21    0    0
CT        1    0    1    0    0    1    1    0    1    1
NC        0    0    0    0    1    0    0    0    0    0
CP                                                     98
PB                                                     41
TE                                                    139
```

Figure 7.5 Sample Results of the Robust Scheduling Method (RSM).

A forward pass is used to compute Early Start (ES) and Early Finish (EF). A backward pass is used to compute Late Start (LS) and Late Finish (LF). Results include Total Slack (TS), Free Slack (FS) as well as a task indicator of Critical Task (CT) and Non Critical (NC). Finally the RSM computations include Critical Path (CP) duration, Project Buffer (PB) and overall expected project duration (TE). The RSM computations are completed for each project independently.

The Project Valuation Method (PVM) implemented in procedure 200 computes the expected delay (DE) for each project, resulting expected value (VE) and value lost (VL). The expected value (VE) is used to compute expected net present value (NPVE) and net present value lost (NPVL) for each project. The potential net present value (NPVP) is also computed. The sum total potential net present value and expected net present value for all projects is computed along with the Program Schedule Efficiency (PSE) for the program. An example of PVM results for one project is shown in Figure 7.6.

```
200 - PROJECT VALUATION METHOD (PVM)

Project                                   1
VP                                  $17,000,000
TP                                       90
LP                                      400
TE                                      139
DE                                       49
VE                                  $11,262,713
VL                                   $5,737,288
NPVP                                $16,336,724
NPVE                                $10,798,944
NPVL                                 $5,537,780
Rate                                  0.000137

TOTAL
NPVE     $86,755,429
NPVP     $109,270,539
PSE      0.7940
```

Figure 7.6 Sample results of the Project Valuation Method (PVM).

The Project Prioritization Method (PPM) implemented in procedure 300 computes Project Productivity Metric (PPM) and Productivity Project Index (PPI) based on productivity analysis in 5.5. The PPM also computes a NPV Project Index (NPI) for use in prioritizing projects using NPV criteria. An example of PPM results for six projects is shown in Figure 7.7.

300 - PROJECT PRIORITIZATION METHOD (PPM)

SPI - Sequential Project Index	1	2	3	4	5	6
NPI - NPV Project Index	6	2	1	5	4	3
Project Productivity Index						
Alpha	0.10	0.10	0.10	0.10	0.10	0.10
Beta	0.10	0.10	0.10	0.10	0.10	0.10
DVLDT	106,675	71,006	52,067	124,200	64,600	45,167
MAXDV	124,200	124,200	124,200	124,200	124,200	124,200
GAMMA	0.86	0.57	0.42	1.00	0.52	0.36
PPM	88,689	112,689	135,027	140,813	104,726	86,424
PPI - Productivity Project Index	5	3	2	1	4	6

Figure 7.7 Sample results of the Project Prioritization Method (PPM).

The Sequential Scheduling Method (SSM) implemented in procedure 500 provides a complete solution with resource assignments and schedules. The solution is generated following the Sequential Task First (STF) rule. The program processes resources in two primary passes as described in 6.3. The first pass assigns resources without changing the end date of the project thus producing a Time Constrained Solutions (TCS). If any project lacks sufficient resources at the end of the first pass it is reinitialized for a second pass. The second pass incrementally applies a lag on tasks until a suitable resource can be assigned thus imposing a delay on the task start and finish dates and to the project.

The Critical Scheduling Method (CSM) implemented in procedure 600 provides a complete solution with resource assignments and schedules. The solution is generated following the Critical Task First (CTF) rule. The program processes resources in three primary passes as described in 6.4. The first pass assigns resources without changing the end date of the project thus producing a Time Constrained Solutions (TCS). If any project lacks sufficient resources at the end of the first pass then resource from a lower priority partially assigned project are redeployed to a higher priority project on a second pass. If some projects are still not completely resource a third pass incrementally applies a lag on tasks until a suitable resource can be assigned thus imposing a delay on the task start and finish dates and to the project.

The Early Scheduling Method (ESM) implemented in procedure 700 provides a complete solution with resource assignments and schedules. The solution is generated following the Earliest Task First (ETF) rule. The program processes resources in two primary passes as described in 6.5. The first pass assigns resources without changing the end date of the project thus producing a Time Constrained Solutions (TCS). If any project lacks sufficient resources at the end of the first pass then those projects are reinitialized and the second pass incrementally applies a lag until a suitable resource can be assigned thus imposing a delay on the task start and finish dates and to the project.

The results for all scheduling procedures (500, 600, 700, 800, 900, 1000) may differ but all are presented in the same format. Key results for each solution include:

- Overall project schedules including buffer (only projects that are completely resourced).
- Start and finish date for each task.
- Resource assigned to each task.
- Skill level assigned to each task.
- Program schedule efficiency.

An example of the format of the assignment matrix (A_{mj}) for a Time Constrained Solution (TCS) is shown in Figure 7.8. By inspection of the example shown only two projects, three (3) and five (5), are fully resourced within the time constraints established by the RSM. All other projects cannot be fully resourced within time constraints and therefore are incomplete and not scheduled. The TCS includes complete scheduling information, valuation and program metrics (PSE) for those projects scheduled. The performance of this TCS will improve as additional resource sets with increased resource numbers are applied.

```
500 - SEQUENTIAL SCHEDULING METHOD(SSM)
TIME CONSTRAINED SOLUTION (TCS)
Project Prioritization PP:          1
Resource Prioritization PR:         1
Resource Quantity MR:              20

                        Assignment Matrix AMJ - Task
Project       1    2    3    4    5    6    7    8    9   10
1             0    3    0    5   15   17    1    9   12    0
2             0   20    0   10   11   14   10    9    6    0
3             0   13   16    4    5    8    5    2    6    0
4             0    0    0    8   18    0   15   12    6    0
5             0    3    7    1    4    1    4    2    2    0
6             0    0    0    0    8   11    1   19    2    0
```

Figure 7.8 Sample Time Constrained Solution (TCS) assignment matrix.

An example of the format of the assignment matrix (A_{mj}) developed in the second pass for a Resource Constrained Solution (RCS) is shown in Figure 7.9. The second pass resets any project not fully assigned in the first pass and reassigns those projects. Reassignment is executed with the same prioritization criteria, by relaxing time constraints and by applying lags and delays to tasks and projects. Upon completion of the second pass all projects are fully resourced and scheduled.

```
500 - SEQUENTIAL SCHEDULING METHOD (SSM)
RESOURCE CONSTRAINED SOLUTION (RCS)
Project Prioritization PP:        1
Resource Prioritization PR:       1
Resource Quantity MR:            20

                  Assignment Matrix AMJ - Task
Project      1    2    3    4    5    6    7    8    9   10
1            0   13   16    5   11   14    4    6    6    0
2            0   20    3   10    1    8   10    2    9    0
3            0   13   16    4    5    8    5    2    6    0
4            0   20    7    8   15   11   15   12    9    0
5            0    3    7    1    4    1    4    2    2    0
6            0   20   13   17    5   18    1   19    2    0
```

Figure 7.9 Sample Resource Constrained Solution (RCS) assignment matrix.

RAMPS results include comprehensive scheduling data for each project as illustrated in Figure 7.10. The updated data includes start (ST) and finish (FN) dates for each task. In order to achieve assignments the computations now includes slack and lag time in the schedule. Computational updates also include critical path (CP), project buffer (PB), expected project duration (TE) and expected delay (DE). The delay is used to update expected net present value (NPVE) for each project. The program results shows an update (for six projects) including total expected net present value and new value for program schedule efficiency (PSE).

```
500 - SEQUENTIAL SCHEDULING METHOD (SSM)
RESOURCE CONSTRAINED SOLUTION (RCS)

Project   1   1   1   1   1   1   1   1   1    1
Task      1   2   3   4   5   6   7   8   9   10
DM        0  14  28  21  14  28  21  35  21    0
DL        0  21  35  35  28  35  35  49  28    0
FJ        0   1   1   2   2   2   2   3   3    0

ES        0   0   0   0  28  28  56  21  77   98
EF        0  14  28  21  42  56  77  56  98   98
LS        0  28   0  21  42  28  56  42  77   98
LF        0  42  28  42  56  56  77  77  98   98
ST        0  22  22   1  50  50  78  22  99  119
FN        0  35  49  21  63  77  98  56 119  119

CT        1   0   1   0   0   1   1   0   1    1
TS        0  28   0  21  14   0   0  21   0    0
SC        0  22   0   0   0   0   0   0   0    0
LG        0   0  22   0   0   0   0   0   0    0

AMJ       0  13  16   5  11  14   4   6   6    0
GM        0   1   1   2   2   2   2   3   3    0
HM        0   2   3   1   2   2   1   1   1    0

CP                                           119
PB                                            41
TE                                           160
DE                                            70
NPVE                                  $8,732,444

TOTAL PROGRAM
NPVE      $79,747,719
NPVP      $109,270,539
PSE       0.7298
```

Figure 7.10 Sample schedule results and program results.

The RAMPS program also includes a Gantt chart to serve as a quick visual indicator of the time period resource assignments (A_{jt}). While the start (ST), finish (FN) and expected duration (TE) for each project should be sufficient for planning resource allocation it is helpful to view the Gantt chart in order to verify assignments and schedules. A sample Gantt chart is shown in Figure 7.11

```
500 - SEQUENTIAL SCHEDULING METHOD (SSM)
RESOURCE CONSTRAINED SOLUTION (RCS)

GANTT CHART
Project     3   3   3   3   3   3   3   3   3   3
Task        1   2   3   4   5   6   7   8   9  10
Day
1           0  13  16   4   0   0   0   0   0   0
2           0  13  16   4   0   0   0   0   0   0
3           0  13  16   4   0   0   0   0   0   0
4           0  13  16   4   0   0   0   0   0   0
5           0  13  16   4   0   0   0   0   0   0
6           0  13  16   4   0   0   0   0   0   0
7           0  13  16   4   0   0   0   0   0   0
8           0  13  16   4   0   0   0   0   0   0
9           0  13  16   4   0   0   0   0   0   0
10          0  13  16   4   0   0   0   0   0   0
11          0  13  16   4   0   0   0   0   0   0
12          0  13  16   4   0   0   0   0   0   0
13          0  13  16   4   0   0   0   0   0   0
14          0  13  16   4   0   0   0   0   0   0
15          0  13  16   4   0   0   0   0   0   0
16          0  13  16   4   0   0   0   0   0   0
17          0  13  16   4   0   0   0   0   0   0
18          0  13  16   4   0   0   0   0   0   0
19          0  13  16   4   0   0   0   0   0   0
20          0  13  16   4   0   0   0   0   0   0
21          0  13  16   4   0   0   0   0   0   0
22          0   0   0   0   5   8   0   2   0   0
23          0   0   0   0   5   8   0   2   0   0
24          0   0   0   0   5   8   0   2   0   0
25          0   0   0   0   5   8   0   2   0   0
26          0   0   0   0   5   8   0   2   0   0
27          0   0   0   0   5   8   0   2   0   0
28          0   0   0   0   5   8   0   2   0   0
29          0   0   0   0   5   8   0   2   0   0
30          0   0   0   0   5   8   0   2   0   0
31          0   0   0   0   5   8   0   2   0   0
32          0   0   0   0   5   8   0   2   0   0
33          0   0   0   0   5   8   0   2   0   0
34          0   0   0   0   5   8   0   2   0   0
35          0   0   0   0   5   8   0   2   0   0
36          0   0   0   0   0   8   5   2   0   0
37          0   0   0   0   0   8   5   2   0   0
38          0   0   0   0   0   8   5   2   0   0
39          0   0   0   0   0   8   5   2   0   0
40          0   0   0   0   0   8   5   2   0   0
```

Figure 7.11 Sample Gantt chart schedule results.

Scheduling and assignment results are generated for all the scheduling procedures (500, 600, 700, 800, 900 and 1000). The DST provides additional useful data for some of the procedures. The Critical Scheduling Method (CSM) in procedure 600 is a three pass algorithm and a Critically Constrained Solution (CCS) is generated as an interim result. The Early Scheduling Method (ESM) in procedure 700 generates and provides a list of tasks that are presorted and ranked in order of earliest task first. The Skill Level Assignment Method (SLAM) in procedures 900 and 1000 provide results on the number of trial solutions and on the number of successful substitutions.

7.2 Experimental Objectives

The experimental objects consist of testing and evaluating the performance of the resource assignment decision model using practical data sets. The RAMPS decision tool is used throughout to provide all computational results. Parameters used in the computational analysis are initialized as follows; nominal value for work rate (q_1 and q_3) is assumed to be 0.3, prioritization parameters (α and β) assumed to be 0.1 and work force shaping parameters (κ and λ) assumed to be zero. Objectives include:

1. Evaluate the effectiveness of the project and resource prioritization methods by comparing Productivity ranking and NPV ranking and by comparing performance with and without resource skill level prioritization. (Results in section 7.5.1)

2. Evaluate sensitivity of the model to productivity prioritization weighting factors α and β reflecting relative confidence (optimistic/pessimistic) of executing the overall program per planned schedules. Productivity prioritization parameters (α and β) will be allows to vary between 0 and 1 while holding other parameters constant. (Results in section 7.5.2)

3. Evaluate and compare the Time Constrained Solution (TCS) and the Resource Constrained Solutions using the constructive priority rule heuristics such as STF, CTF and ETF. (Results in section 7.5.3)

4. Compare the overall Skill Level Assignment Method (SLAM) solution with the Level Constrained Solution (LCS) solution.
(Results in section 7.5.4)

5. Evaluate the Skill Level Assignment Method (SLAM) sensitivity to work rate to characterize how skill level differences and respective differences in work rates will impact the program. Objective three (3) allows the work rate parameters (q_1 and q_2) to vary from 0.1 to 0.5 while holding other parameters constant. (Results in section 7.5.5)

6. Evaluate sensitivity of the model to variation in potential launch date (T_P) by allowing the launch date TP to vary ± 20% while holding other parameters constant. (Results in section 7.5.6)

7. Evaluate sensitivity of the model to resource mix by introducing variation to resource type and skill level. Work force shaping parameters (κ and λ) will be fixed at specific distributions representing opposite mix of work force while retaining other parameters constant. (Results in section 7.5.7)

7.3 Data Set Generation

The experimental objects consist of testing and evaluating the performance of the resource assignment decision model using practical data sets. The data sets while practical are designed to provide non-dominated cases where the methodology can discern priorities regarding projects and assignments. The data sets are grouped into small (J10), medium (J30) and large (J60) project representation. As the size of the project increases so does the number of different types of resource required and the duration of tasks and projects. The complexity of test case networks (ratio of arcs to tasks) is approximately 2 throughout all networks.

A "J10" project data set represents a set of six (6) projects each having ten (10) tasks and using three (3) skill types. The J10 data set represents smaller NPD projects with typically shortest durations. Overall duration for the J10 projects is typically three (3) to six (6) months. The J10 projects are fairly simple examples drawn from standard textbooks (Kerzner 2003, Badiru 1995). J10 data was the primary test case used during the development of the RAMPS program. Complete J10 project data is provided in Appendix B.

The "J30" data set was developed by modeling typical NPD projects across several industries. Case studies (Pu 2009) were drawn from the following industries:

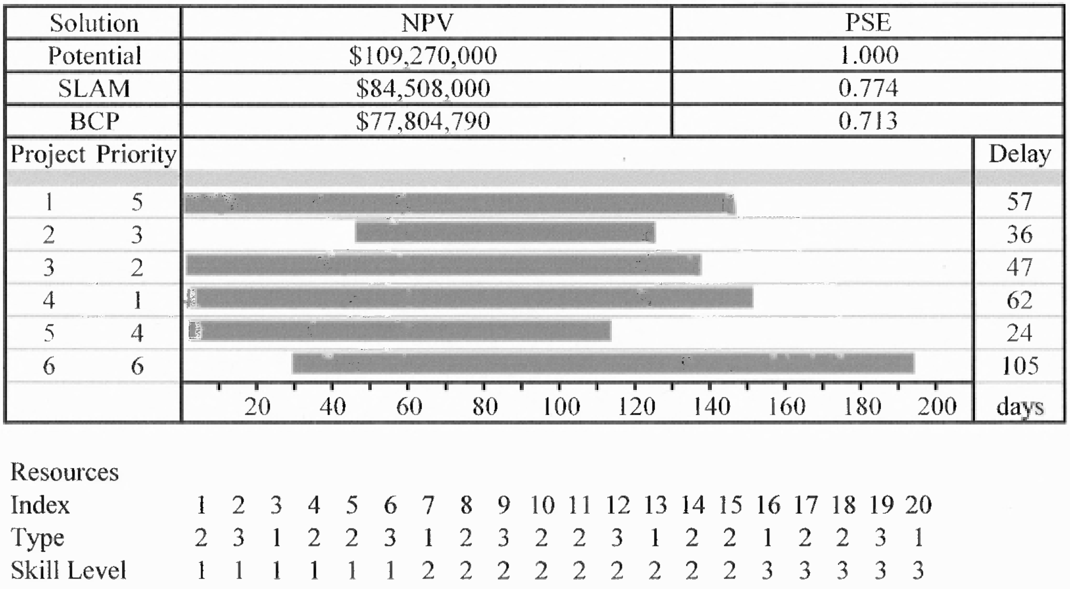
1. Consumer Electronics (LCD TV, iPod)
2. Home Appliance (Refrigerator)
3. Small Appliance (Hair Dryer)
4. Computer Products (Notebook computer)
5. Telecommunications/IT (Routers)

The projects were planned with best estimates of task and durations for versions of these products being introduced rapidly to replace previous generations. Projects were tailored to fit the computation requirements of this dissertation model. Each of the six (6) project networks have exactly thirty (30) tasks and require four (4) resource types including mechanical engineers, electrical engineers, system engineers and manufacturing engineers. The J30 networks represent medium to large NPD projects with overall durations of six (6) to twelve (12) months. The networks diagrams for the J30 data set are provided in Appendix B.

The "J60" data set is comprised of larger networks consisting of four (4) projects each having sixty (60) tasks and requiring five (5) resource types. The J60 projects represent larger projects with duration estimates of twelve (12) to eighteen (18) months. These NPD projects would typically reflect new to the world product development or a fundamental change to a product architecture. The J60 project networks are also found in Appendix B.

7.4 Illustrative Example of Results with J10

The overall solution results contain extensive details on schedules, assignments and metrics. The scheduling information includes start and end date for each project. The Gantt chart in Figure 7.11 summarizes the schedule for six (6) projects in the J10 project data set. Results reflect the project priority and delay (D) beyond the potential launch date (T_p). The potential NPV (NPV_P) and expected NPV (NPV_E) are provided along with the PSE metric for both the SLAM solution and the Best Current Practice (BCP) solution. The nominal case of twenty (20) resources used in the assignments is also shown along with resource type and skill level.



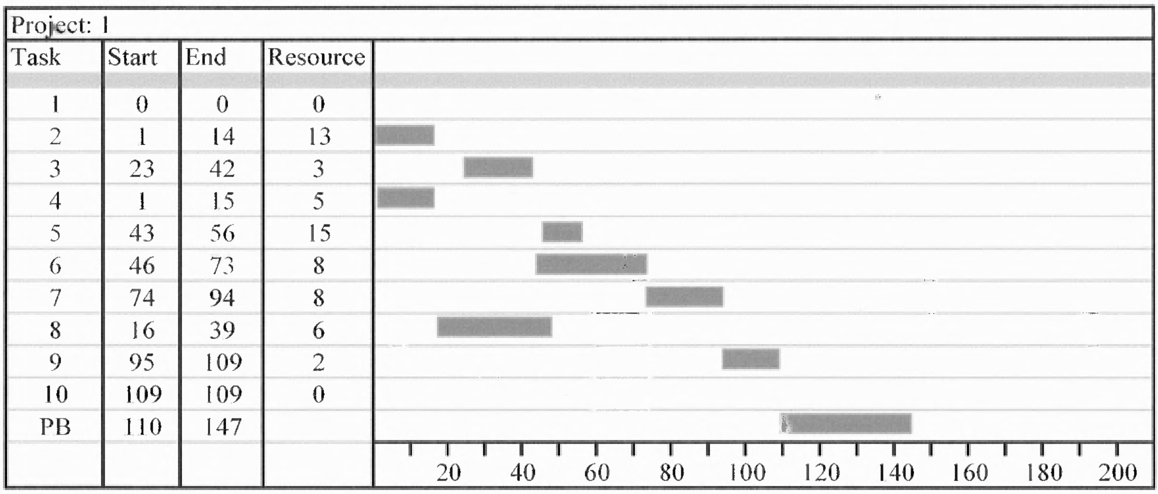
Solution	NPV	PSE
Potential	$109,270,000	1.000
SLAM	$84,508,000	0.774
BCP	$77,804,790	0.713

Project	Priority	Delay
1	5	57
2	3	36
3	2	47
4	1	62
5	4	24
6	6	105

Resources																				
Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Type	2	3	1	2	2	3	1	2	3	2	2	3	1	2	2	1	2	2	3	1
Skill Level	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3

Figure 7.11 Program results summary for J10.

The results include detail schedules for each project. An example for one of the J10 projects is shown in Gantt chart format in Figure 7.12. Data available include start and finish date for each task and resources assigned to each task. In addition the overall schedule includes the Project Buffer (PB) to account for variation in duration estimates.

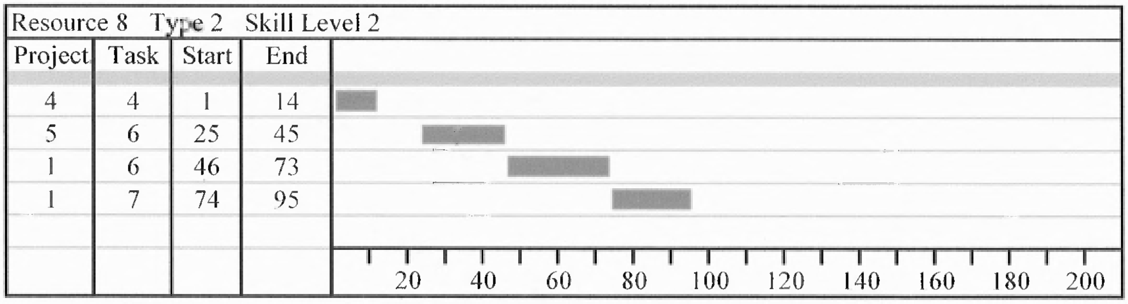


Project: 1

Task	Start	End	Resource
1	0	0	0
2	1	14	13
3	23	42	3
4	1	15	5
5	43	56	15
6	46	73	8
7	74	94	8
8	16	39	6
9	95	109	2
10	109	109	0
PB	110	147	

Figure 7.12 Gantt chart of one J10 project.

The summary information provides assignment and scheduling information for each resource in the inventory. Figure 7.13 provides an example of the resource scheduling across three (3) different projects and four (4) tasks. Resource type and skill level are also referenced.



Resource 8 Type 2 Skill Level 2

Project	Task	Start	End
4	4	1	14
5	6	25	45
1	6	46	73
1	7	74	95

Figure 7.13 Utilization summary for one resource on J10 projects.

7.5 Experimental Results

7.5.1 Prioritization Method Effectiveness

The project and resource prioritization method effectiveness was evaluated by comparing Productivity Project Index (PPI) ranking and NPV project index (NPI) project ranking and by comparing performance with and without resource prioritization by skill level. Results are shown in Figure 7.12 and summarized in Table 7.2.

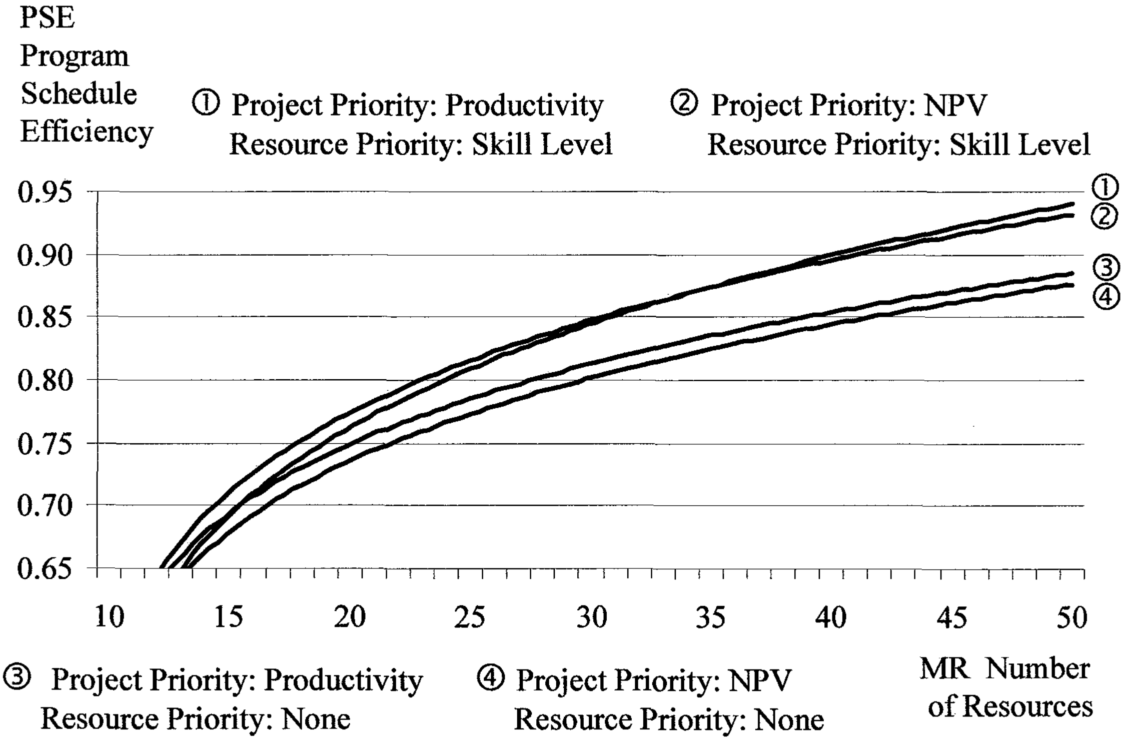


Figure 7.14 Comparison of prioritization methods with J10 data.

The data sets in Figure 7.14 are clearly segmented into two sets with the two (2) curves using resource prioritization clearly out performing those with no resource prioritization. Prioritizing resources by skill level can typically improve the performance of the SLAM solution between 3% and 5% depending on the level of resources applied. Furthermore if resources are not prioritized the algorithm has to work harder. Figure 7.15 quantifies the number of substitutions associated with a SLAM solution using prioritized resources and that of a SLAM solution without prioritized resources. Thus having resources prioritized by skill level allows the algorithm to process more efficiently and results in improved schedule performance over not prioritizing resources.

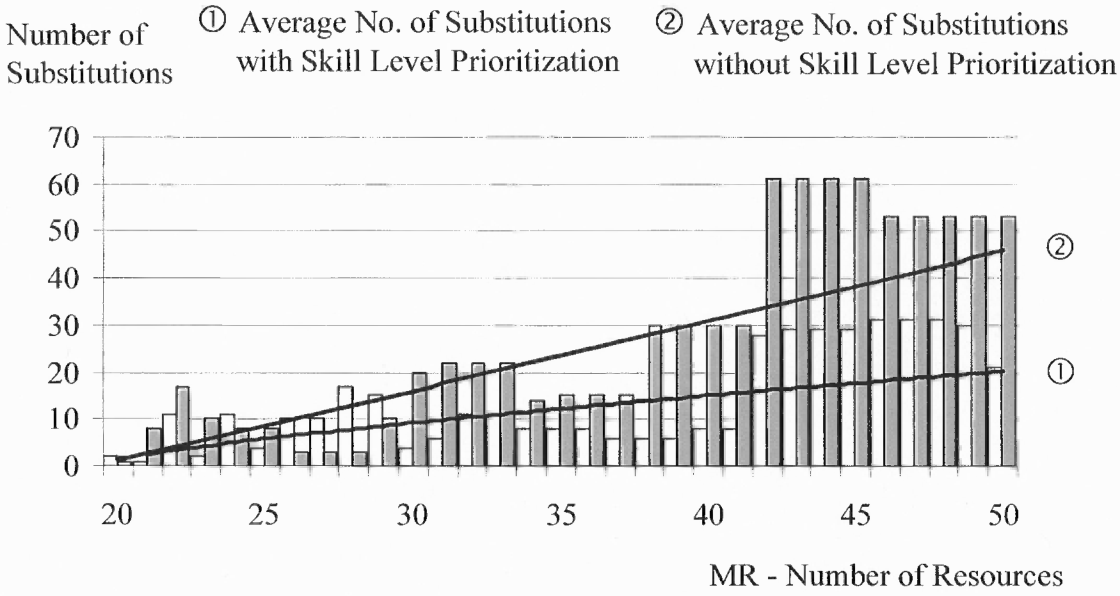


Figure 7.15 Comparison of average number of substitutions.

Detailed evaluation of the data indicates an advantage to prioritizing projects by productivity and to prioritizing resources by skill level as indicated in Table 7.1 thus productivity and skill level prioritization is carried forward in this research.

Table 7.1 Prioritization effectiveness with J10 data.

Project Prioritization	Resource Prioritization	Average Relative PSE Performance	Average Number of Substitutions/ Trial Solutions	CPU Time
Productivity	Skill Level	1.00	10/174	140s
NPV	Skill Level	0.98	10/210	200s
Productivity	None	0.96	18/320	260s
NPV	None	0.94	21/370	300s

7.5.2 Sensitivity to Prioritization Parameters α and β

The PPM ranks projects for scheduling by incorporating productivity prioritization parameters (α and β) reflecting relative confidence (optimistic/pessimistic) of executing the overall program per planned schedules. Productivity prioritization parameters (α and β) will be allows to vary between 0 and 1 while holding other assumptions the same as Objective one (1). A range of solution performance is achieved and the maximum and minimum are plotted in Figure 7.16.

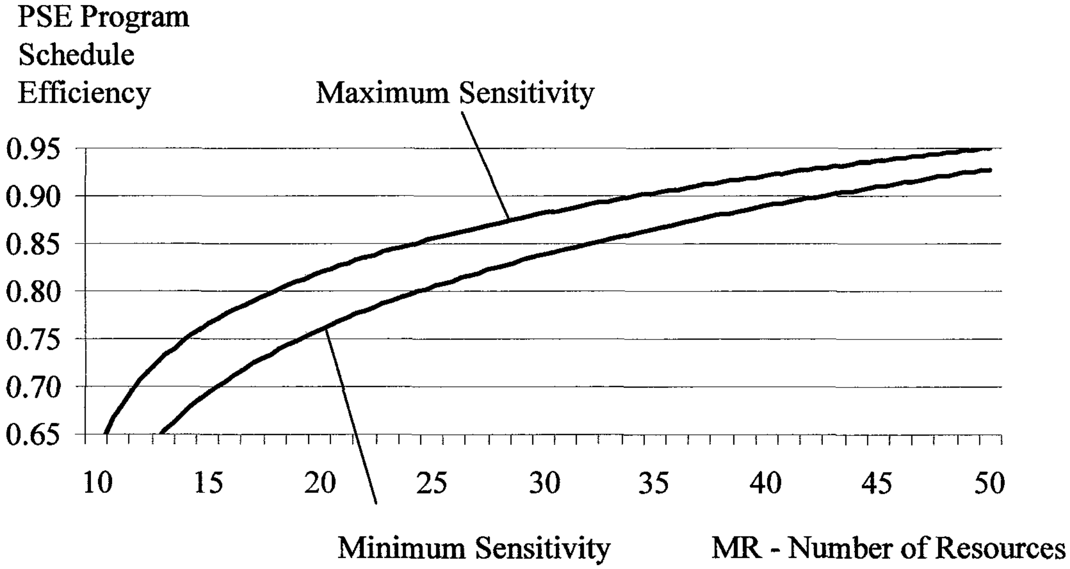


Figure 7.16 Sensitivity of productivity prioritization weighting factors α and β.

The maximum sensitivity tends to occur with higher α and β combinations or when α plus β > 1. The minimum sensitivity tends to occur with lower combinations of α and β combinations or when α plus β < 1.

7.5.3 Comparison of Constructive Solution Methods

The Best Constructive Solution (BCS) is formulated by selecting the solution from constructive solution with the best PSE performance at a particular resource level. The constructive methods considered include the Sequential Scheduling Method (SSM), Critical Scheduling Method (CSM) and Early Scheduling Method (ESM). A comparison of relative performance at various resource levels is shown in Figure 7.16. The performance of Time Constrained Solutions and Resource Constrained Solutions are plotted for a range of resource levels from ten (10) to fifty (5). The resource cost baseline is also shown in the diagram for reference in comparing the rate of change of performance for the incremental cost added.

The Time Constrained Solutions (TCS) shown in Figure 7.17 reflect resource assignments and scheduling that can be achieved if the dates of tasks and projects is inflexible. All of the TCS show a consistent performance that results in projects completed in a stepwise function as resources are increased. All TCS for all the methods result in lower performance than RCS except at the very highest resource levels where the TCS and RCS become equal. Where resources are scarce the SSM logically schedules the most important project first and additional projects are scheduled to completion at a faster rate. Thus the SSM yields the best TCS performance where resources are constrained. Where resources are more plentiful the ESM results in a TCS advantage since completing the earliest tasks first tend to finish projects more quickly.

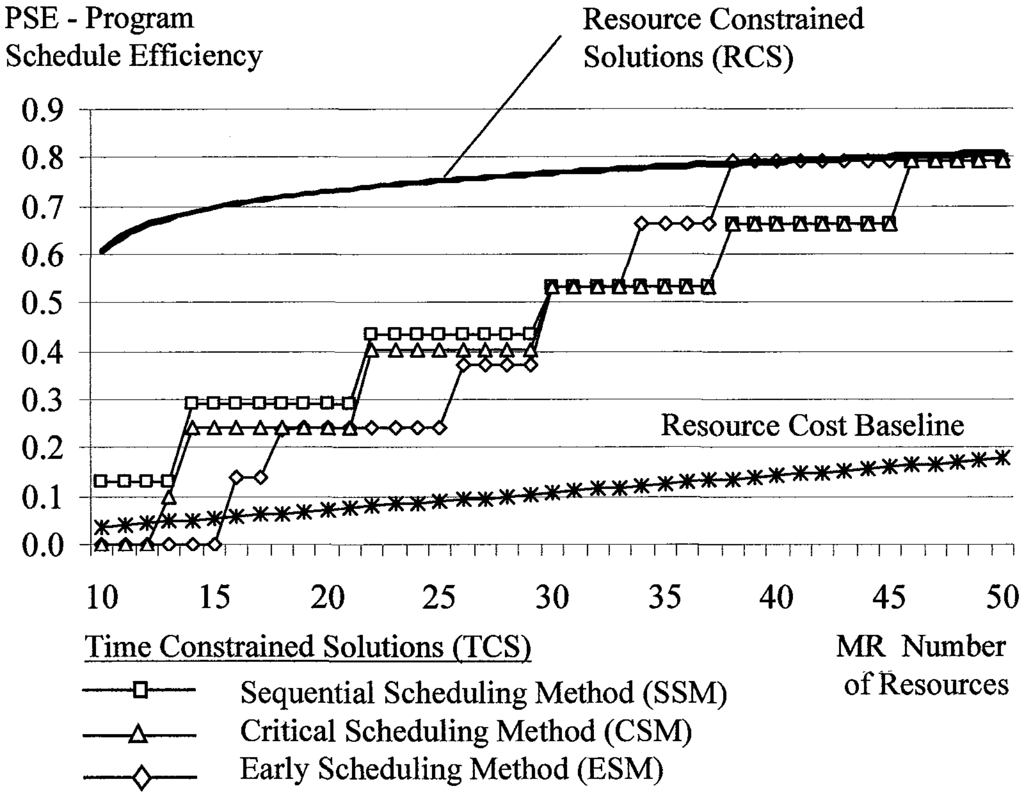


Figure 7.17 Comparison of the constructive scheduling methods.

All of the constructive methods produce a Resource Constrained Solution (RCS) with similar performance as shown in Figure 7.18. At low resource levels (for example below ten (10) resources all solutions revert to the Sequential Scheduling Method (SSM). At some levels the Early Scheduling Method (ESM) solution exceeds the performance of the others. Over the entire range of resources the Critical Scheduling Method (CSM) achieves an average relative performance that offers a slight advantage. If computing time becomes an issue any one of the constructive methods can provide an adequate feasible solution. Note that the purpose of the constructive methods is to establish the Best Constructive Solution (BCS) at each resource level and the RAMPS program can easily run the three methods (or any one of the methods) and save the best performing solution as the BCS.

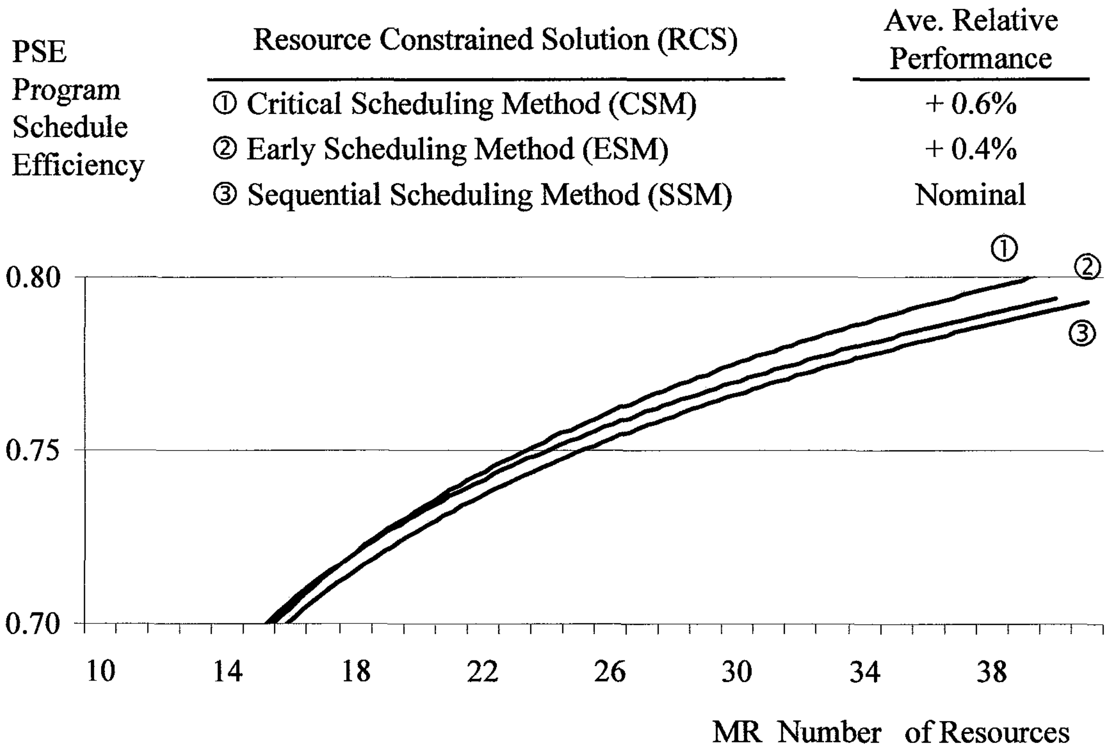


Figure 7.17 Comparison of the constructive methods with differentiation.

7.5.4 Comparison of Improved Solution Methods

The SLAM solution is compared against the LCS solution in this section. The LCS solution is initialized with the Best Constructive Solution and including assignment of skill levels at assigned tasks. The SLAM solution takes advantage of skill level substitution to achieve even higher productivity. The evaluation assumes a nominal work rate (q_1 and q_3) equal to 0.3 and the prioritization parameters (α and β) equal to 0.1. Work force shaping parameters (κ and λ) are assumed to be zero thus all resource sets are generated automatically. The results shown in Figure 7.19 for the J10 data set shows that the SLAM averages 6% higher than the LCS solution across the range evaluated. The

SLAM/LCS solutions for data set J30 and J60 are shown in Figure 7.20 and Figure 7.21 respectively. Results for both J30 and J60 shows that the SLAM solution averages 5% higher than the LCS solution across the range evaluated.

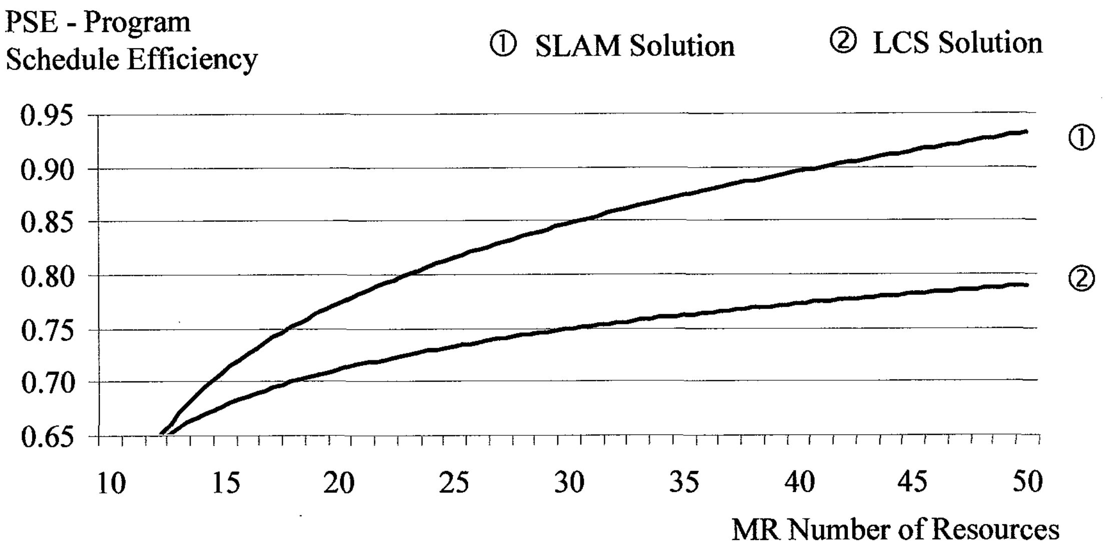


Figure 7.19 Overall solution comparison for the J10 data set.

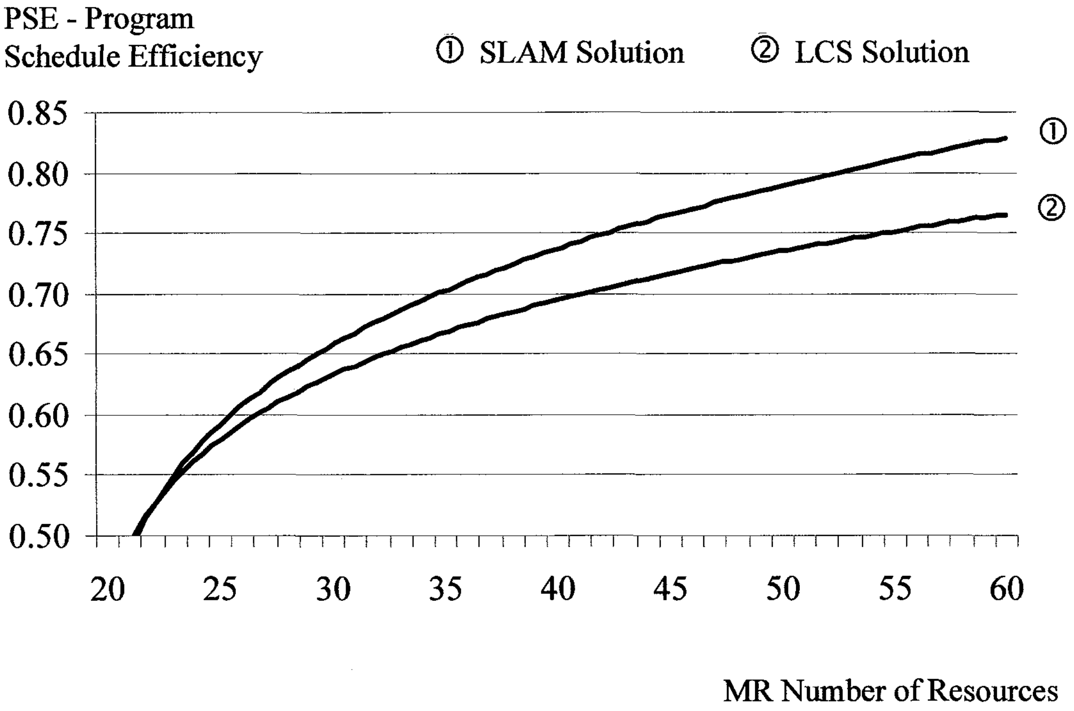


Figure 7.20 Overall solution comparisons for J30 data set.

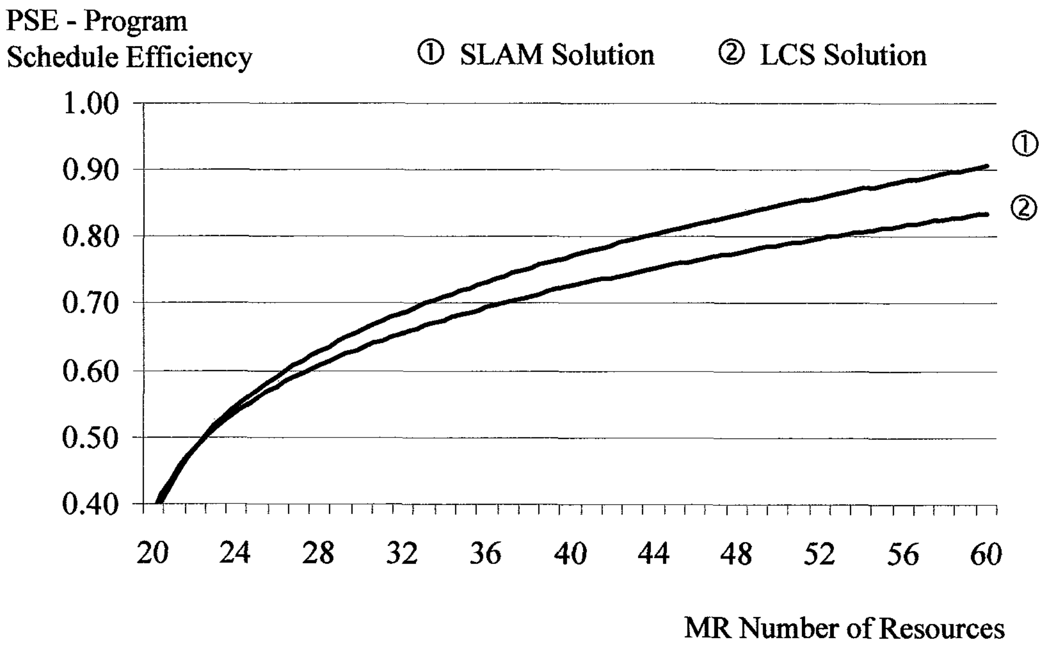


Figure 7.21 Overall solution comparisons for J60 data set.

7.5.5 Sensitivity to Work Rate

The Skill Level Assignment Method (SLAM) solution was evaluated to determine how skill level differences and respective differences in work rates will impact the program performance. Work rate parameters (q_1 and q_2) are allowed to vary from 0.1 to 0.5 while applying other parameters the same. Results for the SLAM solution shown in Figure 7.22 indicate that increasing work rate improves the SLAM solution. In all cases the SLAM solution has improved performance over the BCS solution. The SLAM solution benefits from having faster workers on longer tasks and slower workers on shorter tasks.

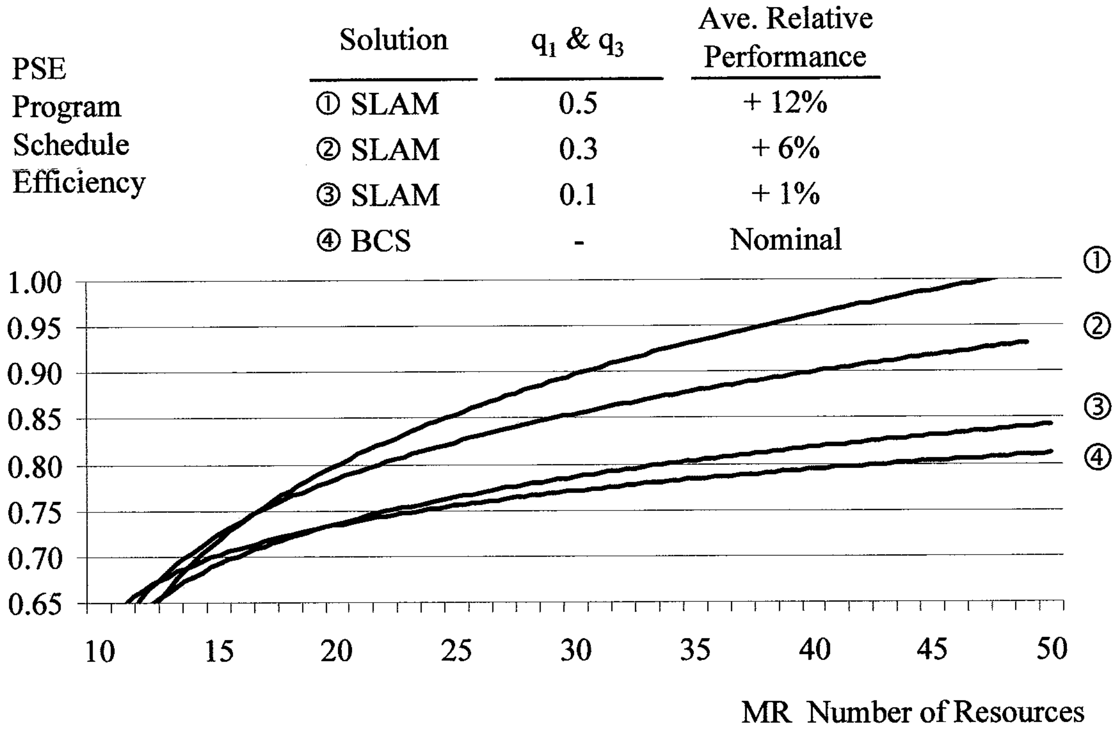


Figure 7.22 Sensitivity to varying work rates.

7.5.6 Sensitivity to Launch Date

Project prioritization and resource scheduling are closely linked to delay relative to potential launch date (T_P). During project planning and execution the potential launch date may be changed to reflect changing market conditions. For instance the customer may want to buy the product sooner or alternatively cannot commit to purchase the product until later than originally planned. The objective is allow the potential launch date TP to vary ± 20% while holding other assumptions constant to evaluate the performance. The launch date sensitivity is shown in Figure 7.23.

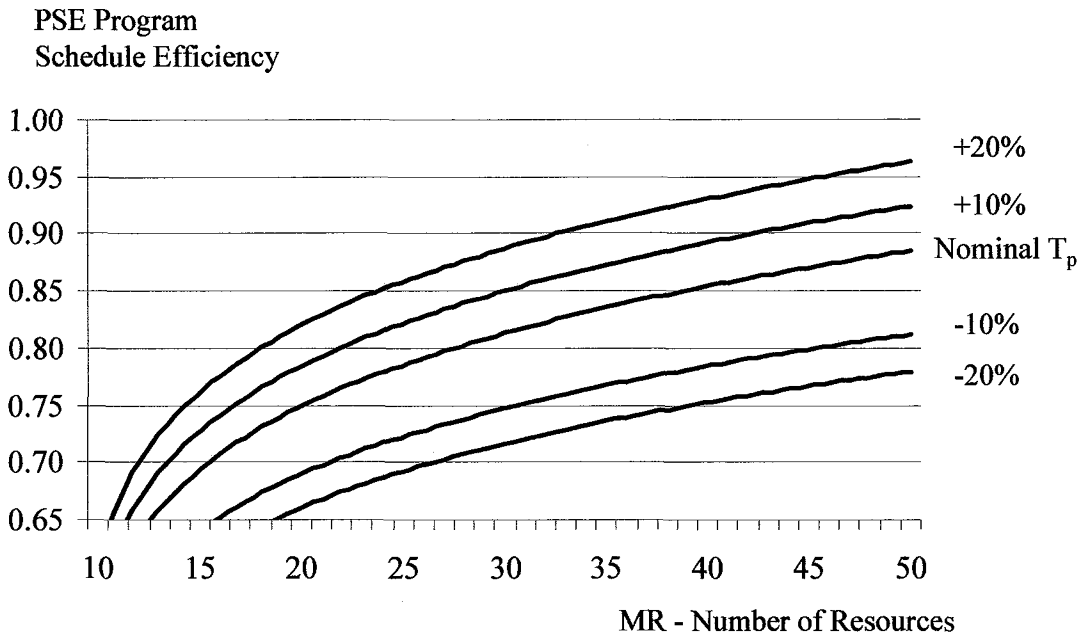


Figure 7.23 Sensitivity to variation in launch date by ± 20%.

7.5.7 Sensitivity to Workforce Mix

The objective is to evaluate the model sensitivity to resource mix by introducing variation to resource type and skill level. Work force shaping parameters (κ and λ) will be fixed at specific distributions representing a mix of work force while retaining other assumptions constant. The description of the mix determined by the resource generator, a strong mix and a weak mix are provided in Table 7.2 and illustrated in Figure 7.24.

Table 7.2 Sensitivity to workforce mix assumptions.

Parameters		Strong Mix	Generator Mix	Weak Mix
Projects Prioritized by Productivity	Type	Matches Project Demand Profile	Matches Demand Profile	Equal Amount, κ1 = 0.2,κ2 = 0.4, κ3 = 0.4
Resources Prioritized by Skill Level	Level	Skewed Higher λ1 = 0.4, λ2 = 0.5 λ3 = 0.1	Uniform Distribution	Skewed Lower, λ1 = 0.1, λ2 = 0.5, λ3 = 0.4

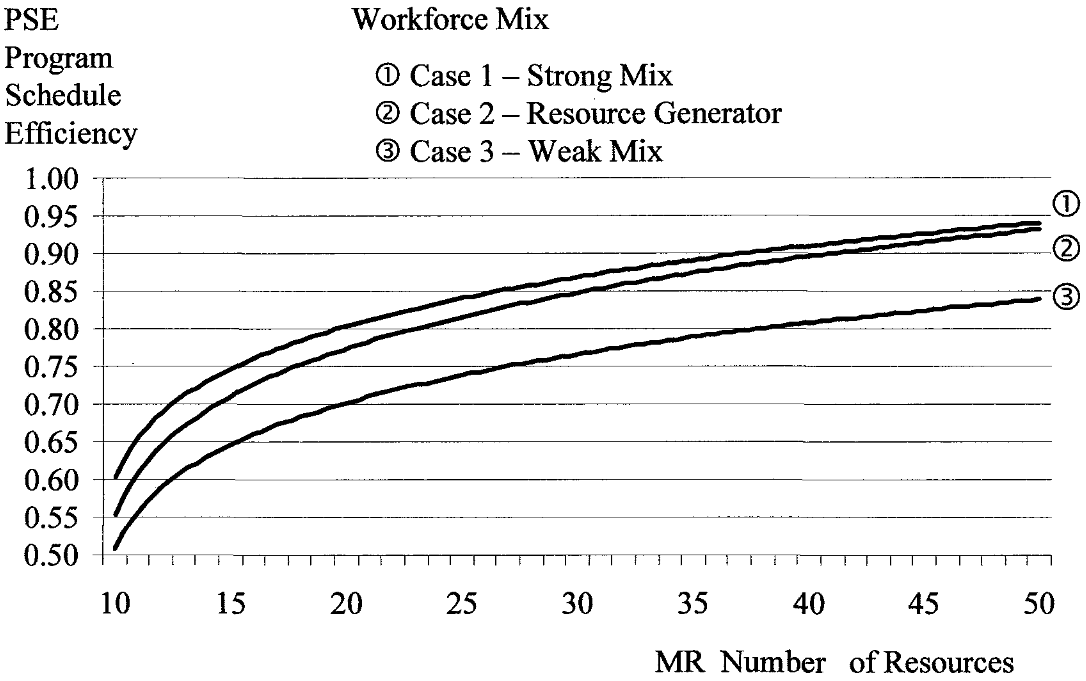


Figure 7.24 Sensitivity to workforce mix.

CHAPTER 8

DISCUSSION AND FUTURE RESEARCH DIRECTION

8.1 Discussion

This dissertation addresses specific aspects of the resource assignment problem crucial to NPD success. Short Life Technology Intensive NPD products and projects are characterized as a class of products having higher technical complexity, require development by a variety of technical disciplines, have relatively short development intervals, in the range of six (6) to eighteen (18) months, and short life spans, in the range of one (1) to three (3) years. Project risk is higher for SLTI projects due to technical complexity and constraints on the availability of technical resources of differentiated discipline and skill level to carry out the development. SLTI projects are launched into stationary market windows where unrecoverable loss occurs with any delay in the launch date. Product development is carried out in multi-project environments where products are developed in parallel and often replaced in rapid succession.

The NPD resource assignment problem formulation matches projects and tasks with available resources in required time periods to achieve product launch objectives. The solution incorporates a Robust Scheduling Method (RSM) including components of the Critical Path (CP) and Critical Chain (CC) method to develop schedules accommodating duration estimate variation in the solutions. A time-sensitive valuation function anchored to product launch date is used in the Project Valuation Method (PVM) to derive a metric that enables comparison of projects. The Project Prioritization Method (PPM) uses the project valuation and develops a productivity index as the basis for prioritizing projects for resource scheduling. The resource assignment and scheduling method includes a Constructive Solution which is a feasible solution formed through iterative search using priority rule heuristics. The Constructive Solution is developed as a feasible solution and as the basis for further improvements based on skill level assignment. The Skill Level Assignment Method (SLAM) solution uses an improvement heuristic that considers productivity gains that may be achieved by interchanging resources of differing skill levels and corresponding work rates.

The Resource Assignment and Multi-Project Scheduling (RAMPS) program is a decision tool developed to solve the resource assignment problem. The RAMPS program automates the entire methodology presented in this dissertation. An experimental analysis using simulated project and resource data sets has demonstrated consistency in the methodology

Observations from this research are:

1. A comparison of project prioritization methods shows improved performance of approximately two percent (2%) for programs prioritized by productivity criteria versus NPV criteria. The data indicates that productivity criteria should be considered when planning multi-project NPD programs.

2. A comparison of resource prioritization methods shows improved performance of approximately four percent (4%) for solutions prioritizing resources by skill level versus solutions where no resource prioritization occurs. This data suggest that significant gains can be achieved through inventorying and prioritizing resource skill levels as a precursor to resource assignment.

3. The Skill Level Assignment Method (SLAM) has the potential to improve assignment solutions approximately five percent (5%) versus solution than have not been optimized for skill level. Most NPD programs will benefit from including a SLAM methodology to improve productivity of project schedules, improve program performance and ensure the best utilization of available skilled resources.

    a. The SLAM algorithm will typically have a greater impact on projects that are not prioritized by productivity, where resources are not prioritized by skill level and on projects having longer tasks where the benefit of interchanging resources of differing skill level is greater.

b. The SLAM algorithm impact will typically have less impact where projects are prioritized by productivity, where resources are prioritized by skill level and on projects having tasks with short duration that have limited productivity benefit from interchanging resources of differing skill levels.

4. The RAMPS decision tool incorporates the entire methodology presented in this dissertation. Computational experiments and analysis indicates that the RAMPS program and embedded algorithms provide excellent solutions to the multi-project NPD resource assignment problem.

8.2 Future Research Direction

Opportunities for future research include:

1. The "simplified" critical chain technique used in the Robust Scheduling Method applies only the project buffer while omitting other aspects of the Critical Chain Project Management (CCPM) methodology. The assignment method may be expanded to include a full blown Critical Chain Project Management including resource leveling and the use of feeder, resource and capacity buffers.

2. The triangular life-cycle valuation function provides an approximate profile of a product revenue life-cycle that declines rapidly without claim to precision. Future enhancements to the Project Valuation Method may incorporate actual life cycle functional relationships conforming more precisely to actual life-cycles revenue or profit profiles.

3. The Project Valuation Method presented in this research emphasizes revenue rather than profit. Future research may incorporate target costing and projected development resource expense as an integral component of the valuation model.

4. The Project Prioritization Method established in this dissertation is primarily based on financial measures with modification of performance of the entire program is accounted for through the weighting factors α and β. Future research may expand the prioritization model to include qualitative factors and risk factors for each project and account for interaction among projects.

5. In formulating a constructive solution there are dozens of priority rules to choose from in order to reach a feasible solution. In this research three priority rules have been selected to formulate the feasible solution. It may be beneficial to investigate other priority rules (such as longest duration first, shortest duration first and others) to determine which ones provide an advantage in establishing the constructive solution.

6. The improvement heuristic solution embodied in the Skill Level Assignment Method may be evolved by investigating metaheuristic algorithms now considered state of the art in solving the single project RCPS problem. Best single project solutions add forward-backward multi-pass techniques to improve schedules constructed with metaheuristics such as Branch and Bound, Genetic Algorithms and Tabu search.

7. Assignment algorithms such as the ones presented in this research allow a resource to be assigned to a task on one project one day and to a task on another project the next day while assuming the same rate of productivity on both projects. In future research it may be beneficial to explore the impact a learning curve that may be representative of moving a resource from one project to another.

8. The methodology in this dissertation was evaluated with project data sets generated to emphasize non-dominated programs and making resource assignment decisions that discern the priorities for resource allocation. As research in multi-project scheduling continues it may be desirable to generate and standardize larger data sets that include standard non-dominated multiple project sets for use in computational study.

9. The RAMPS decision tool can be evolved and improved to provide to allow quicker and simpler execution.

APPENDIX A

FLOW CHARTS

The RAMPS program flow charts are contained in this appendix.

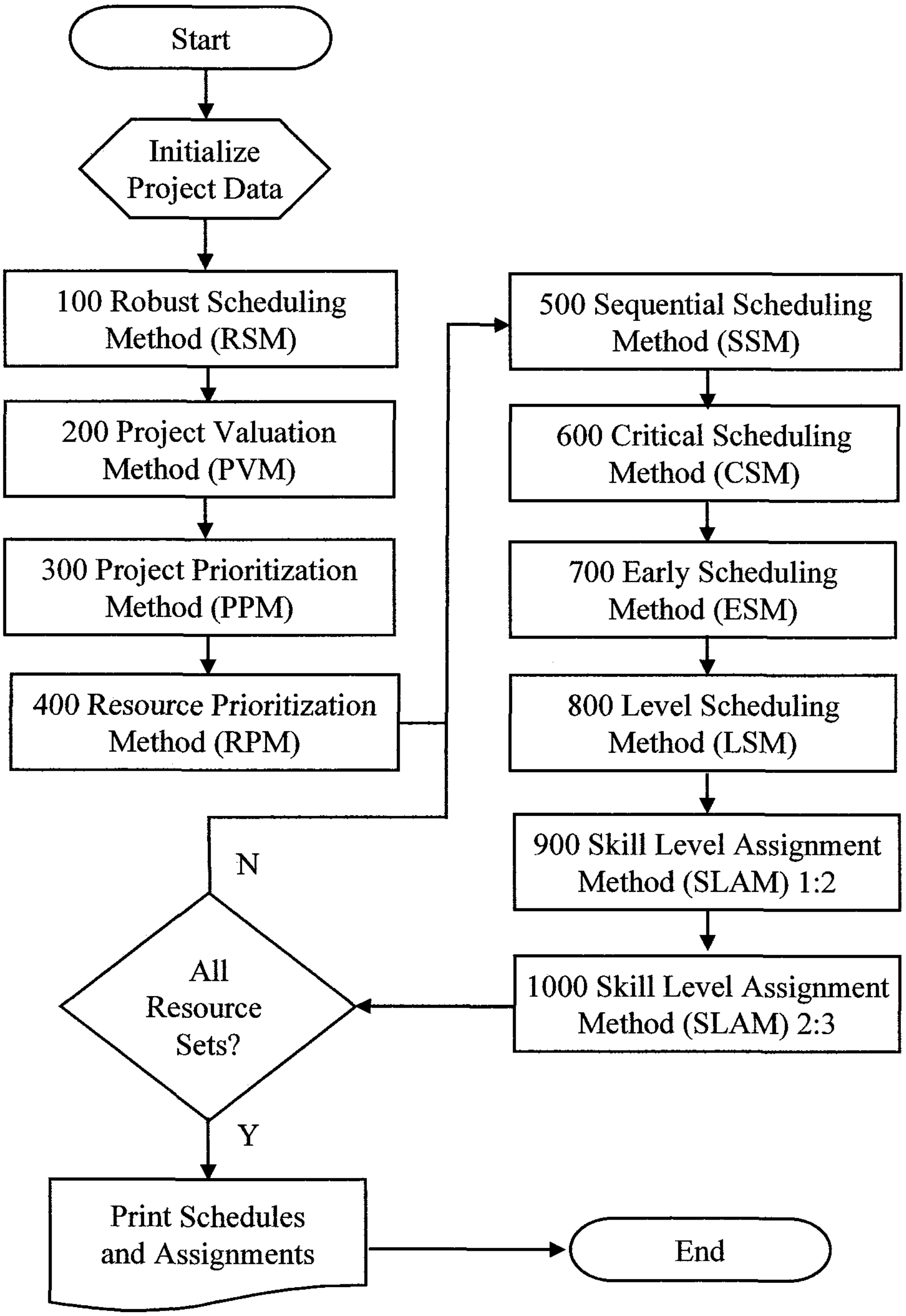

Start
Initialize Project Data
100 Robust Scheduling Method (RSM)
200 Project Valuation Method (PVM)
300 Project Prioritization Method (PPM)
400 Resource Prioritization Method (RPM)
500 Sequential Scheduling Method (SSM)
600 Critical Scheduling Method (CSM)
700 Early Scheduling Method (ESM)
800 Level Scheduling Method (LSM)
900 Skill Level Assignment Method (SLAM) 1:2
1000 Skill Level Assignment Method (SLAM) 2:3
All Resource Sets?
N
Y
Print Schedules and Assignments
End

Figure A.1 Flow Chart - Overall program.

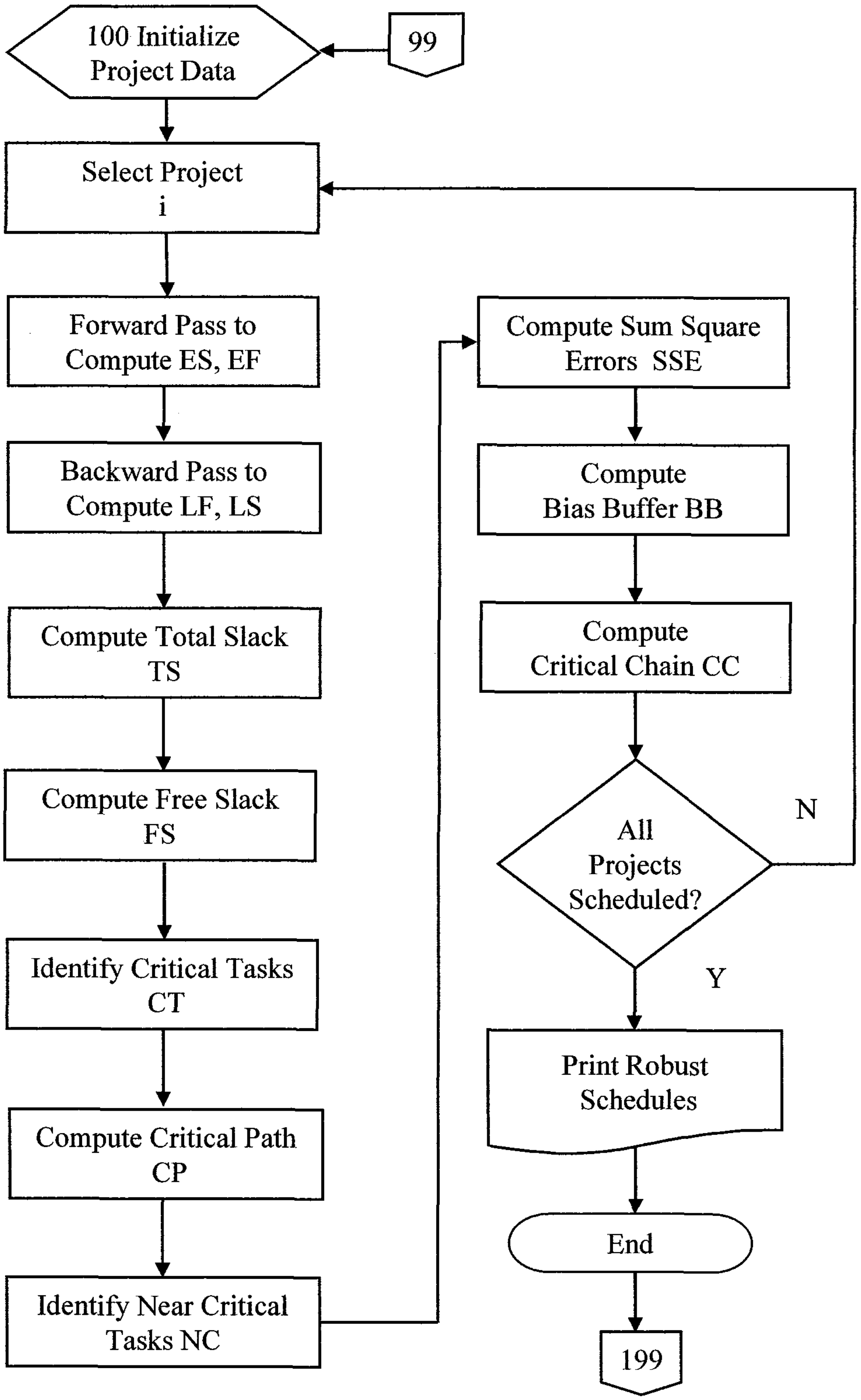

100 Initialize Project Data
99
Select Project i
Forward Pass to Compute ES, EF
Backward Pass to Compute LF, LS
Compute Total Slack TS
Compute Free Slack FS
Identify Critical Tasks CT
Compute Critical Path CP
Identify Near Critical Tasks NC
Compute Sum Square Errors SSE
Compute Bias Buffer BB
Compute Critical Chain CC
All Projects Scheduled?
N
Y
Print Robust Schedules
End
199

Figure A.2 Flow Chart - Robust Scheduling Method (RSM).

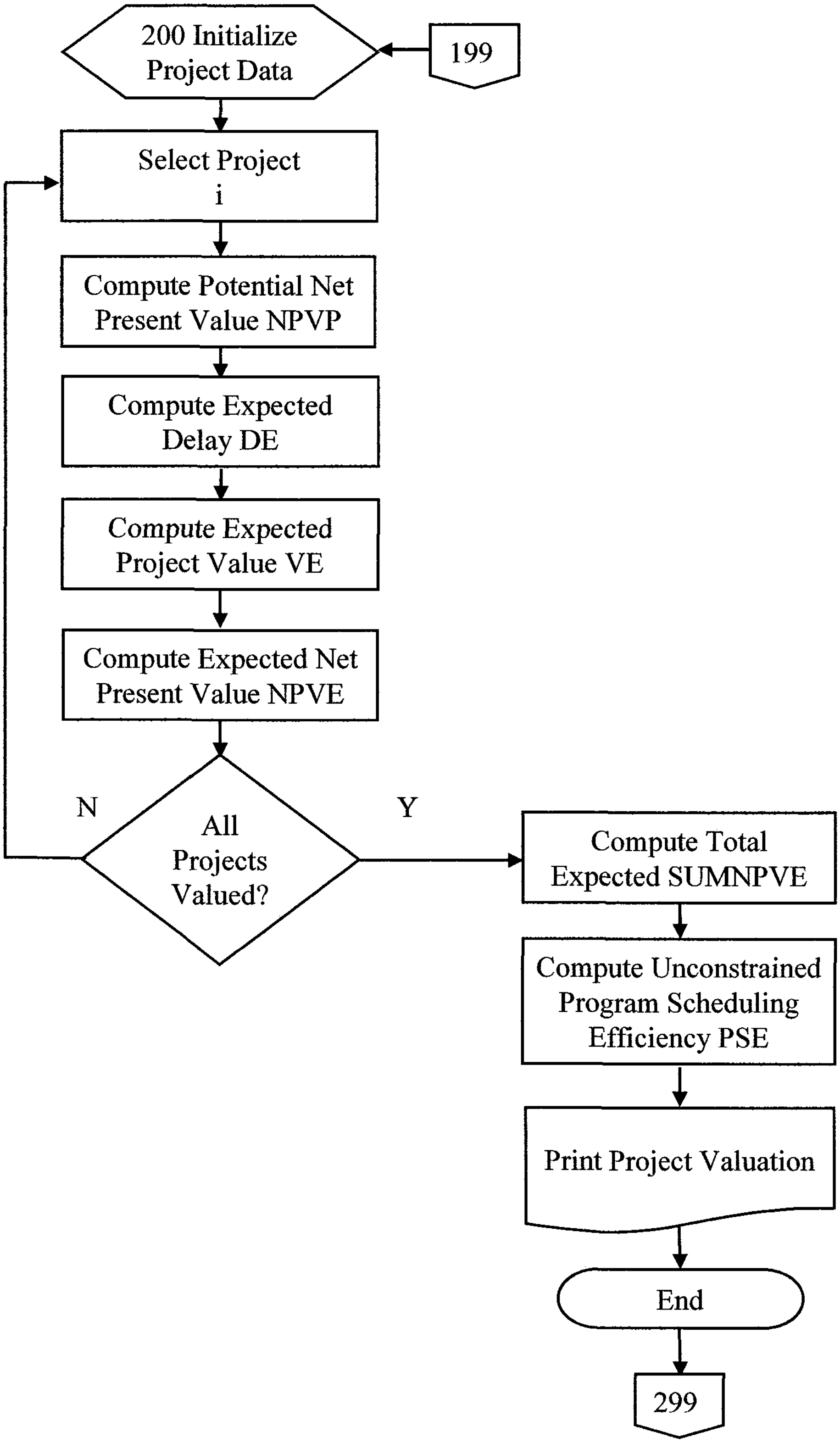

200 Initialize Project Data
199
Select Project i
Compute Potential Net Present Value NPVP
Compute Expected Delay DE
Compute Expected Project Value VE
Compute Expected Net Present Value NPVE
N
All Projects Valued?
Y
Compute Total Expected SUMNPVE
Compute Unconstrained Program Scheduling Efficiency PSE
Print Project Valuation
End
299

Figure A.3 Flow Chart - Project Valuation Method (PVM).

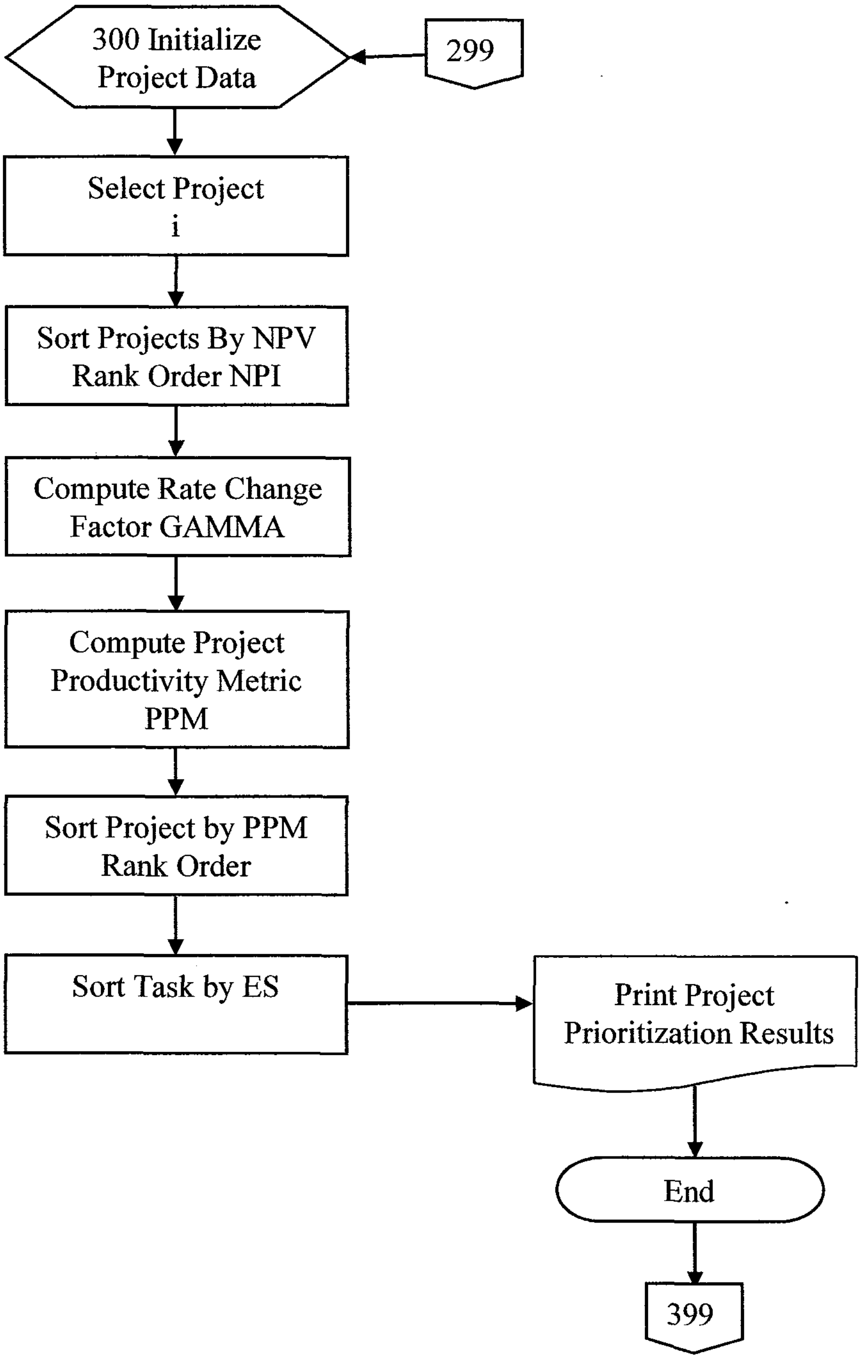

300 Initialize Project Data
299
Select Project i
Sort Projects By NPV Rank Order NPI
Compute Rate Change Factor GAMMA
Compute Project Productivity Metric PPM
Sort Project by PPM Rank Order
Sort Task by ES
Print Project Prioritization Results
End
399

Figure A.4 Flow Chart - Project Prioritization Method (PPM).

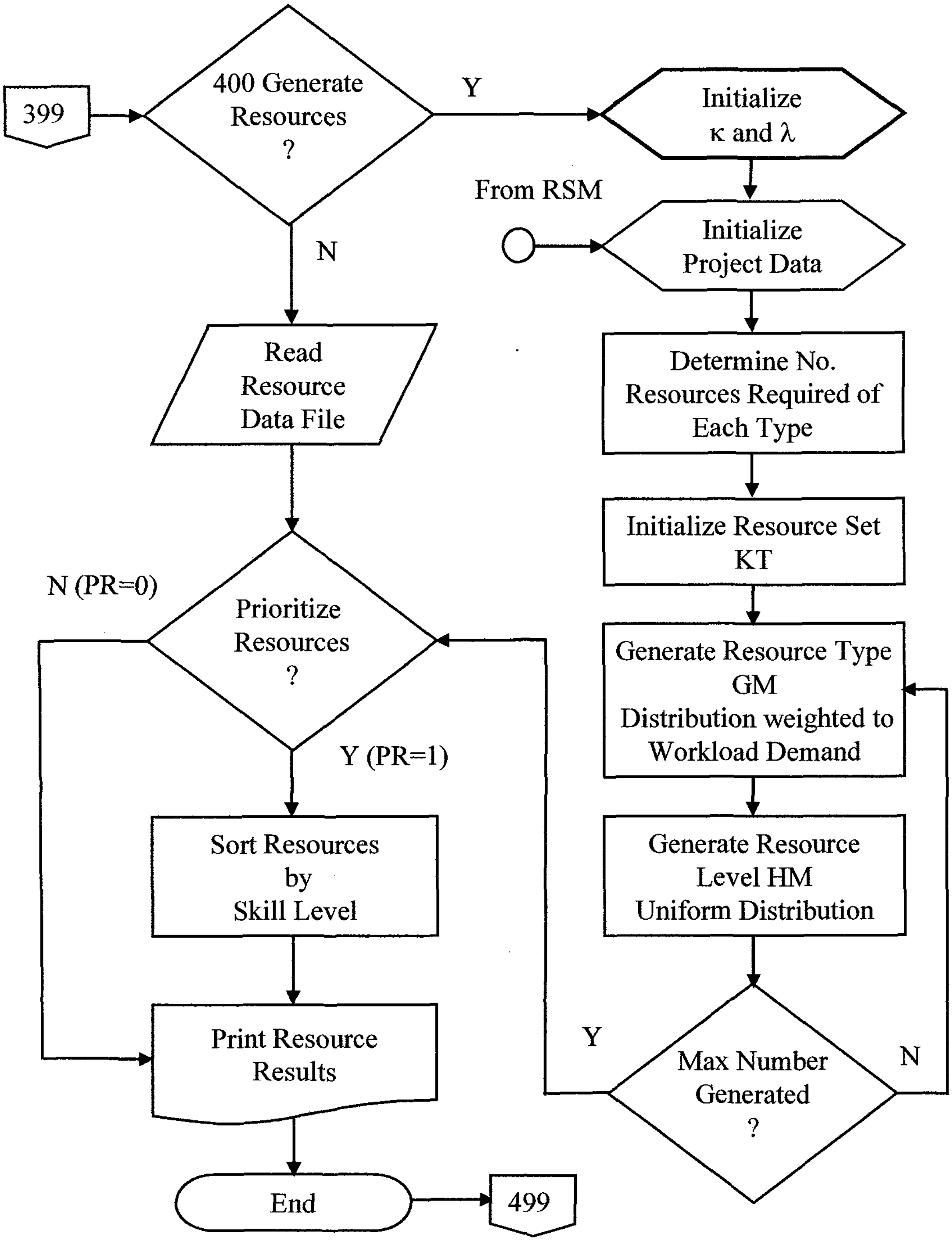

399
400 Generate Resources ?
Y
Initialize κ and λ
From RSM
Initialize Project Data
N
Read Resource Data File
Determine No. Resources Required of Each Type
Initialize Resource Set KT
N (PR=0)
Prioritize Resources ?
Generate Resource Type GM Distribution weighted to Workload Demand
Y (PR=1)
Sort Resources by Skill Level
Generate Resource Level HM Uniform Distribution
Y
Print Resource Results
Max Number Generated ?
N
End
499

Figure A.5 Flow Chart - Resource Prioritization Method (RPM).

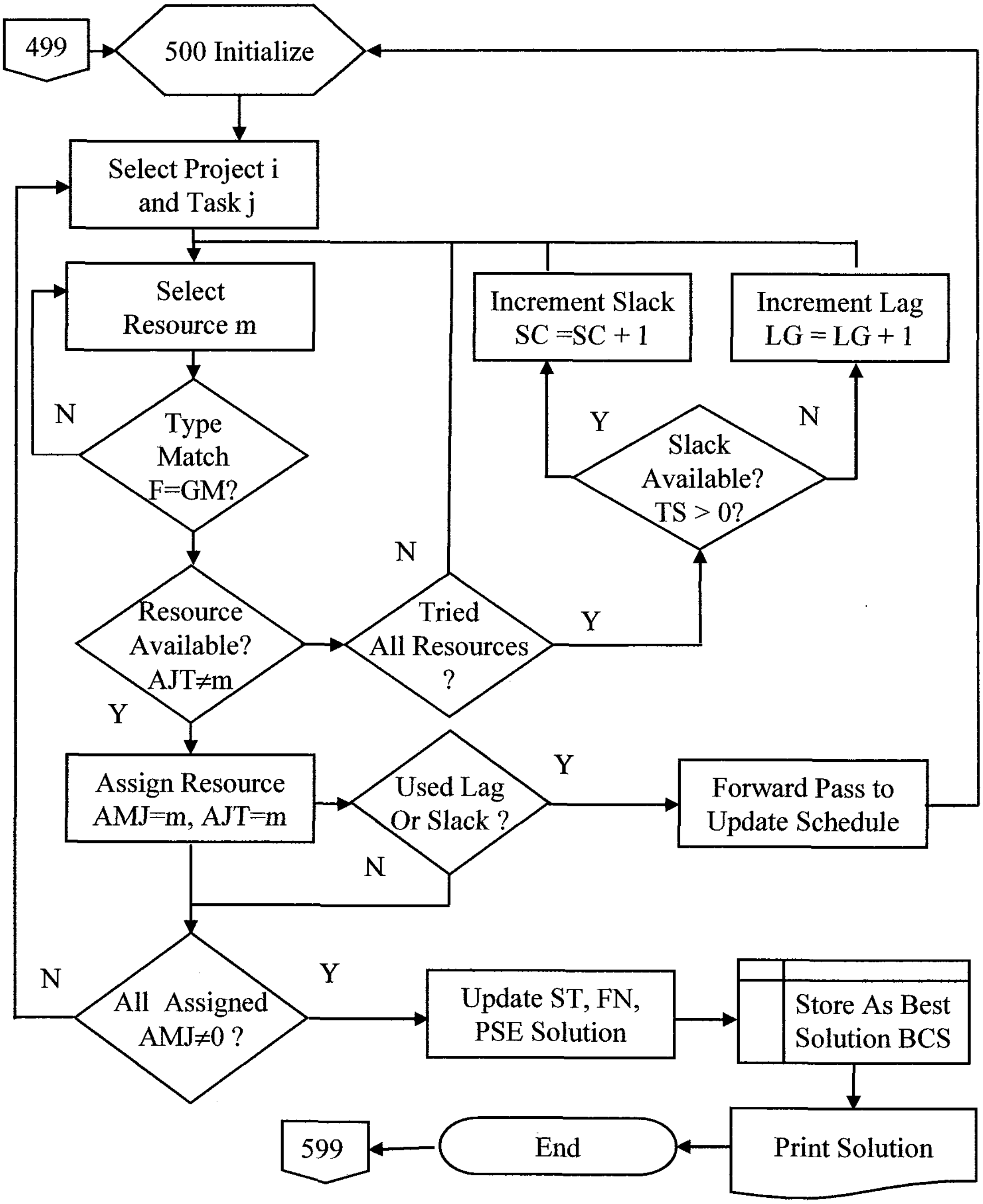

499
500 Initialize
Select Project i and Task j
Select Resource m
Increment Slack SC =SC + 1
Increment Lag LG = LG + 1
N
Type Match F=GM?
Y
Slack Available? TS > 0?
N
N
Resource Available? AJT≠m
Tried All Resources ?
Y
Y
Assign Resource AMJ=m, AJT=m
Used Lag Or Slack ?
Y
Forward Pass to Update Schedule
N
N
All Assigned AMJ≠0 ?
Y
Update ST, FN, PSE Solution
Store As Best Solution BCS
599
End
Print Solution

Figure A.6 Flow Chart - Sequential Scheduling Method (SSM).

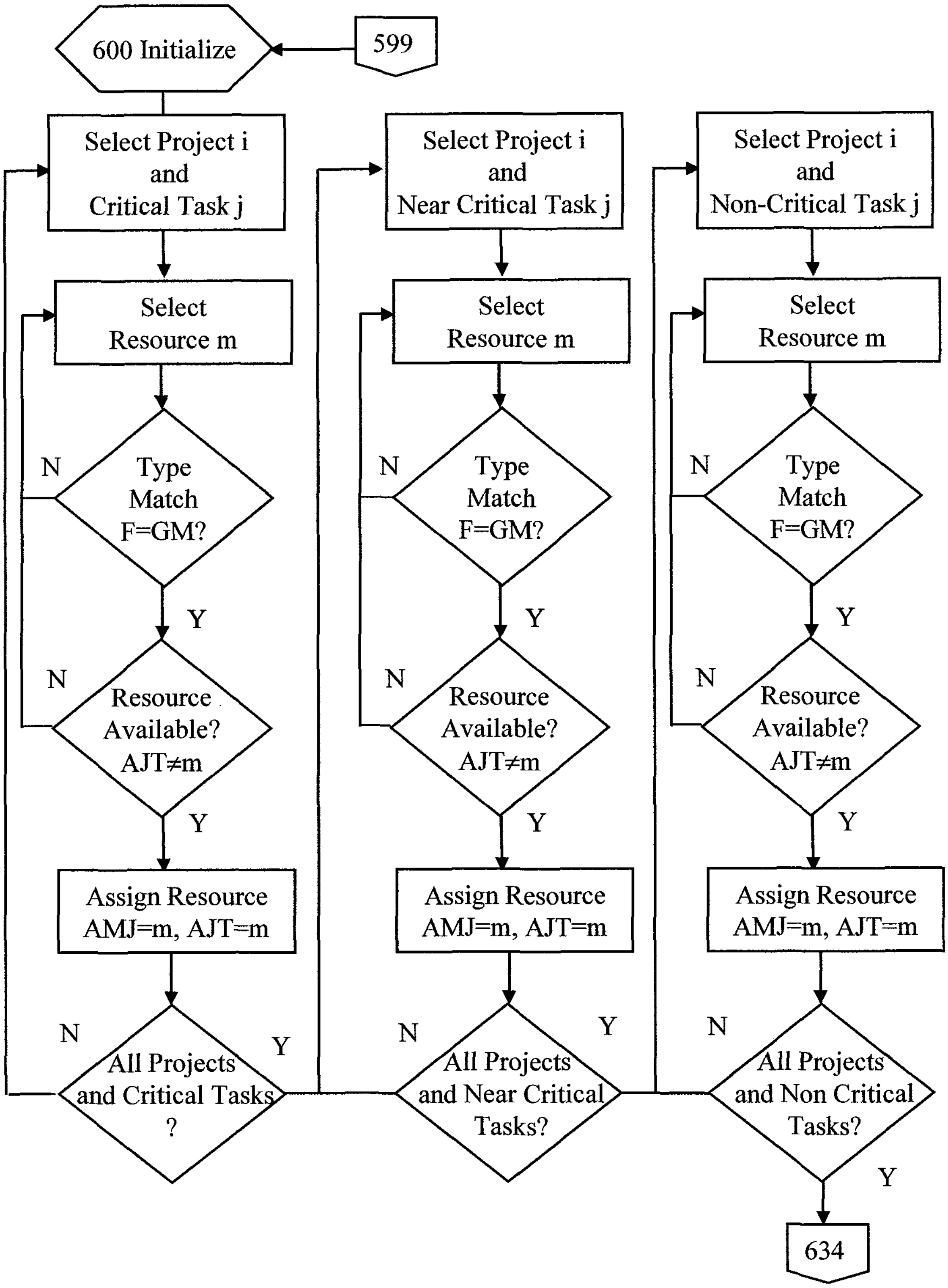

600 Initialize
599
Select Project i and Critical Task j
Select Resource m
Type Match F=GM?
N
Y
Resource Available? AJT≠m
N
Y
Assign Resource AMJ=m, AJT=m
All Projects and Critical Tasks ?
N
Y
Select Project i and Near Critical Task j
Select Resource m
Type Match F=GM?
N
Y
Resource Available? AJT≠m
N
Y
Assign Resource AMJ=m, AJT=m
All Projects and Near Critical Tasks?
N
Y
Select Project i and Non-Critical Task j
Select Resource m
Type Match F=GM?
N
Y
Resource Available? AJT≠m
N
Y
Assign Resource AMJ=m, AJT=m
All Projects and Non Critical Tasks?
N
Y
634

Figure A.7 Flow Chart - Critical Scheduling Method (CSM) .

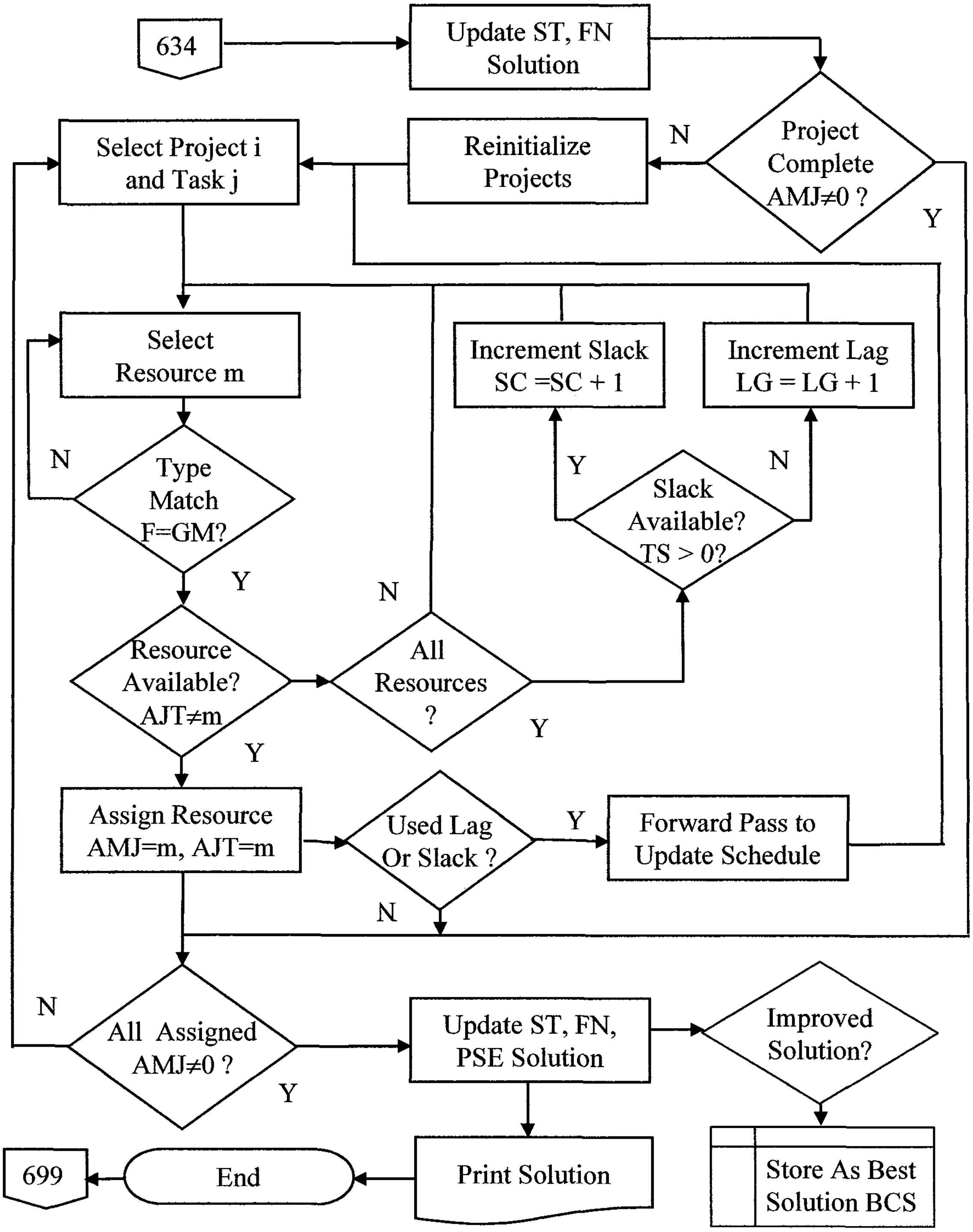


Figure A.8 Flow Chart - Critical Scheduling Method (CSM) - Continued.

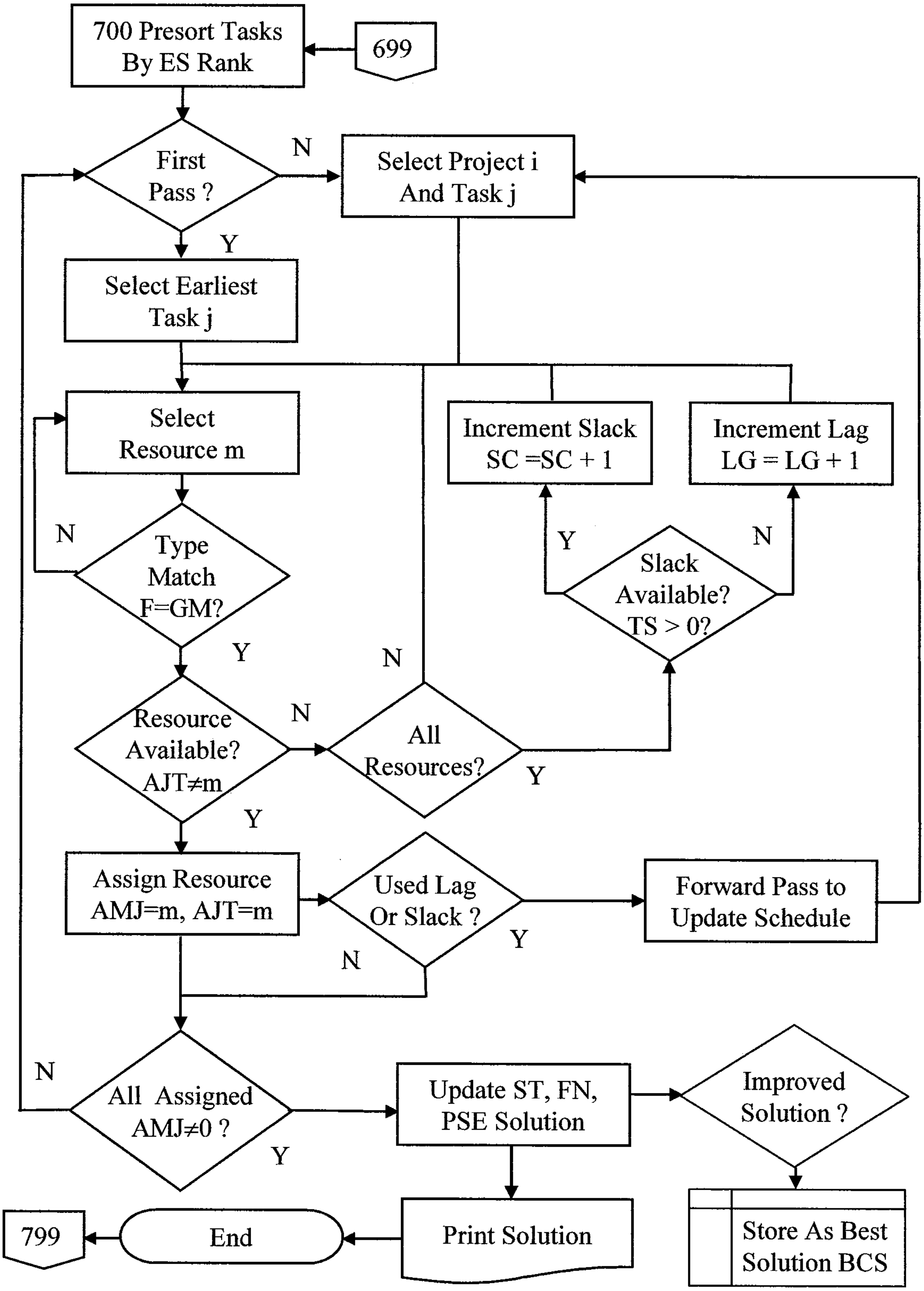

700 Presort Tasks By ES Rank
699
First Pass ?
N
Select Project i And Task j
Y
Select Earliest Task j
Select Resource m
Increment Slack SC =SC + 1
Increment Lag LG = LG + 1
N
Type Match F=GM?
Y
Slack Available? TS > 0?
N
Y
N
Resource Available? AJT≠m
N
All Resources?
Y
Y
Assign Resource AMJ=m, AJT=m
Used Lag Or Slack ?
Y
N
Forward Pass to Update Schedule
N
All Assigned AMJ≠0 ?
Y
Update ST, FN, PSE Solution
Improved Solution ?
799
End
Print Solution
Store As Best Solution BCS

Figure A.9 Flow Chart - Early Scheduling Method (ESM).

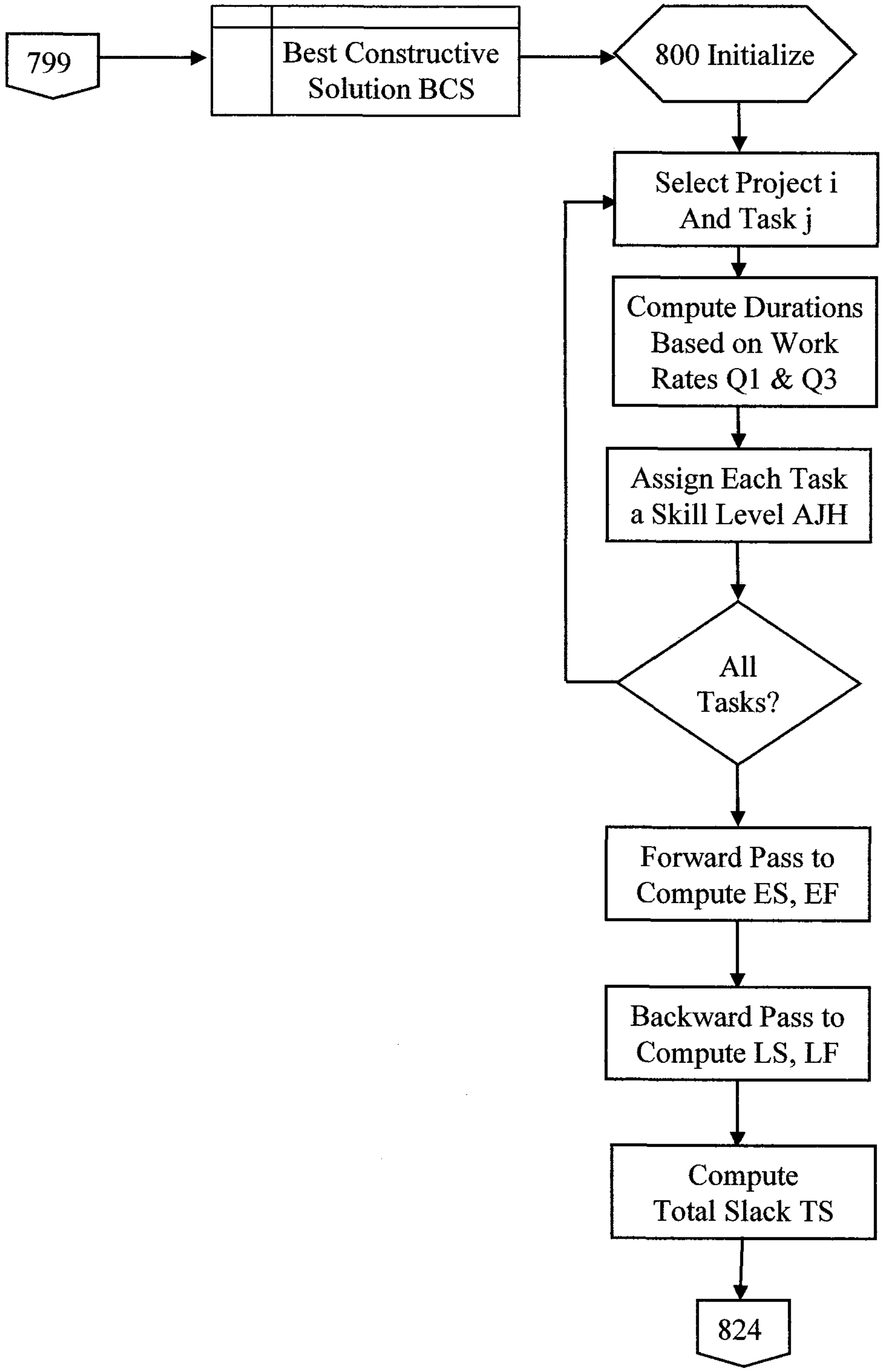

799
Best Constructive Solution BCS
800 Initialize
Select Project i And Task j
Compute Durations Based on Work Rates Q1 & Q3
Assign Each Task a Skill Level AJH
All Tasks?
Forward Pass to Compute ES, EF
Backward Pass to Compute LS, LF
Compute Total Slack TS
824

Figure A.10 Flow Chart - Level Scheduling Method (LSM).

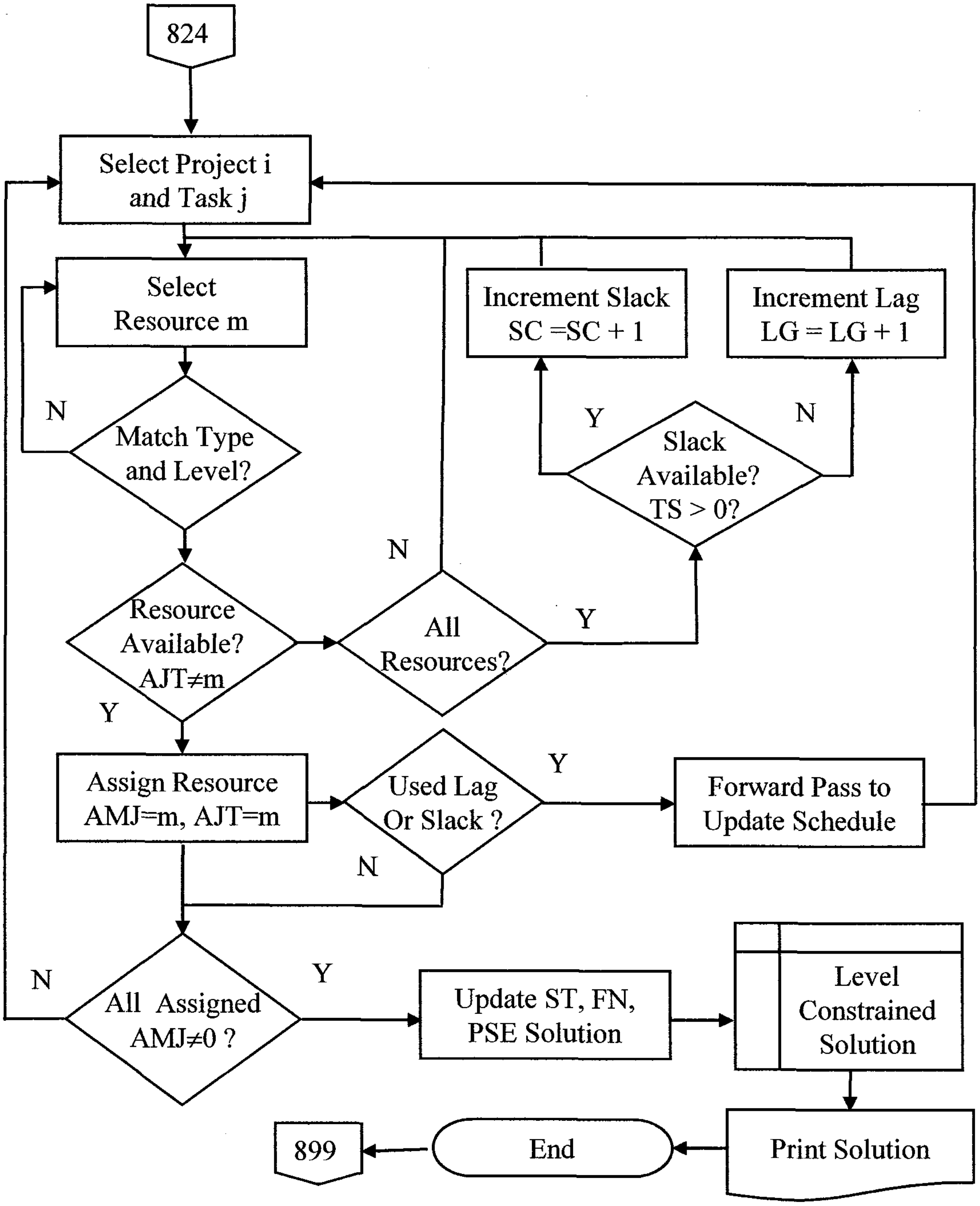

824
Select Project i and Task j
Select Resource m
Increment Slack SC =SC + 1
Increment Lag LG = LG + 1
N
Match Type and Level?
Y
N
Slack Available? TS > 0?
N
Resource Available? AJT≠m
All Resources?
Y
Y
Assign Resource AMJ=m, AJT=m
Used Lag Or Slack ?
Y
Forward Pass to Update Schedule
N
N
All Assigned AMJ≠0 ?
Y
Update ST, FN, PSE Solution
Level Constrained Solution
899
End
Print Solution

Figure A.11 Flow Chart - Level Scheduling Method (LSM) - Continued.

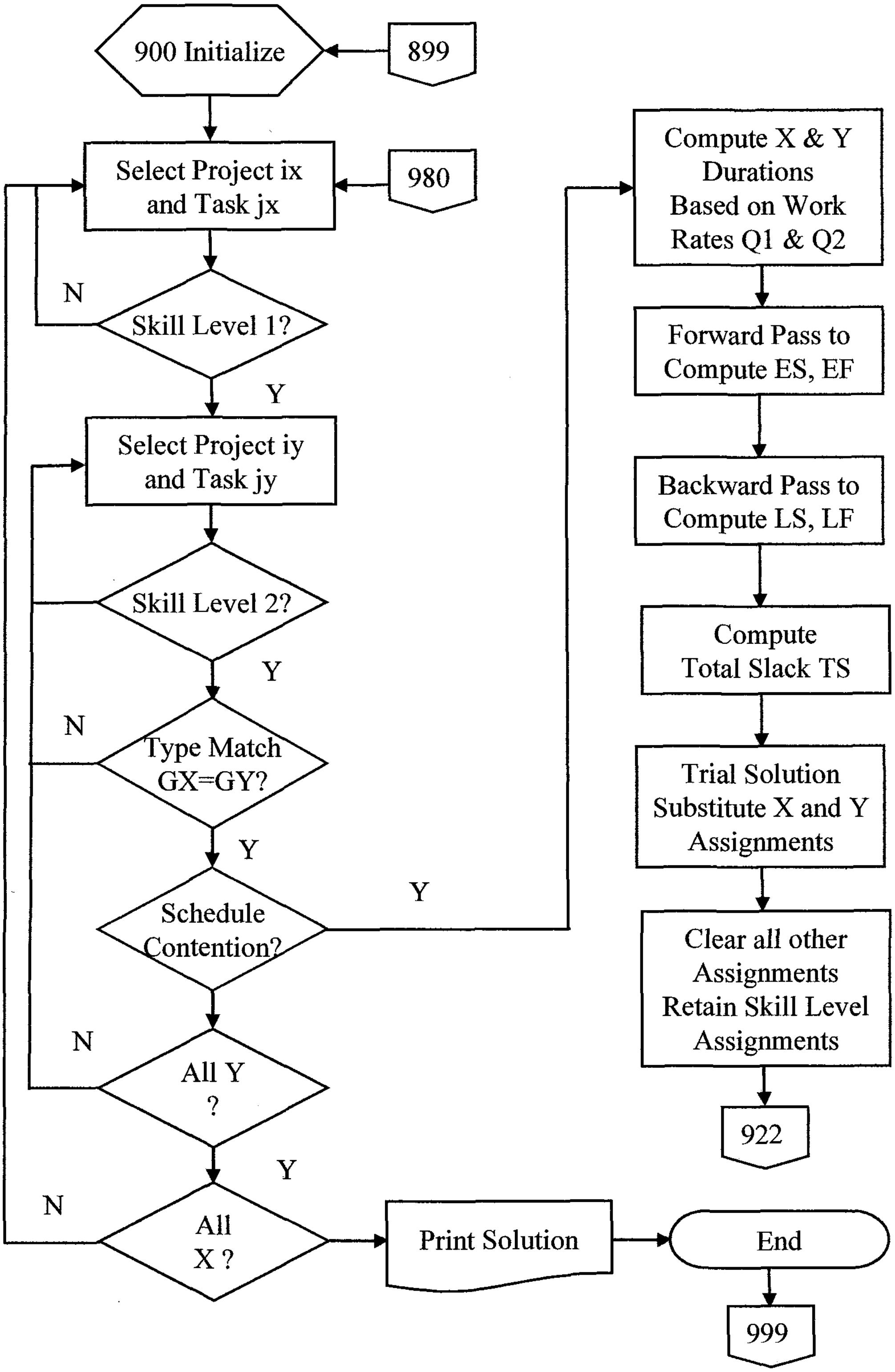

900 Initialize
899
Select Project ix and Task jx
980
N
Skill Level 1?
Y
Select Project iy and Task jy
Skill Level 2?
Y
N
Type Match GX=GY?
Y
Schedule Contention?
Y
N
All Y ?
Y
N
All X ?
Print Solution
End
999
Compute X & Y Durations Based on Work Rates Q1 & Q2
Forward Pass to Compute ES, EF
Backward Pass to Compute LS, LF
Compute Total Slack TS
Trial Solution Substitute X and Y Assignments
Clear all other Assignments Retain Skill Level Assignments
922

Figure A.12 Flow Chart - Skill Level Assignment Method (SLAM) 1:2 .

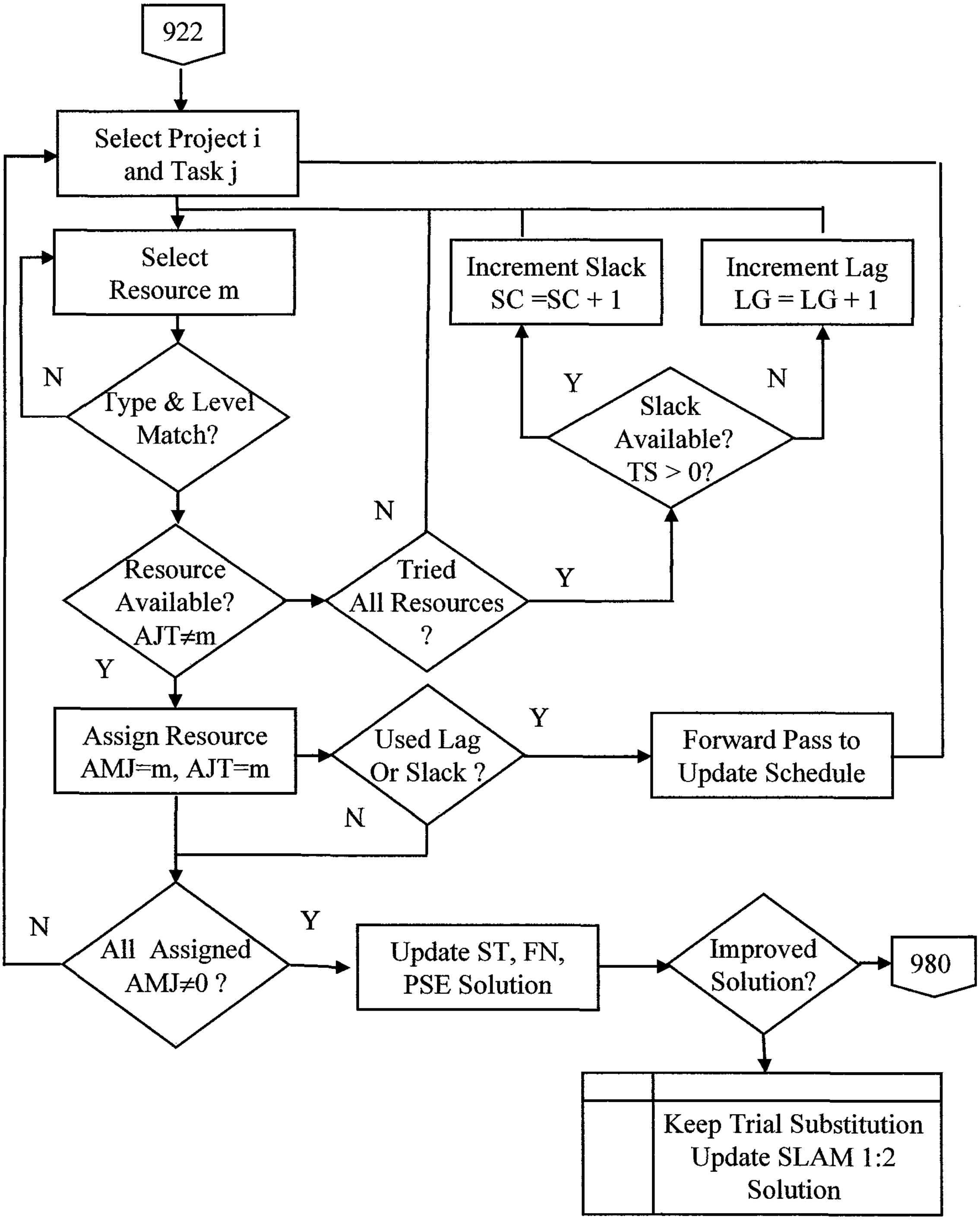

922
Select Project i and Task j
Select Resource m
Increment Slack SC =SC + 1
Increment Lag LG = LG + 1
N
Type & Level Match?
Y
Slack Available? TS > 0?
N
N
Resource Available? AJT≠m
Tried All Resources ?
Y
Y
Assign Resource AMJ=m, AJT=m
Used Lag Or Slack ?
Y
Forward Pass to Update Schedule
N
N
All Assigned AMJ≠0 ?
Y
Update ST, FN, PSE Solution
Improved Solution?
980
Keep Trial Substitution Update SLAM 1:2 Solution

Figure A.13 Flow Chart - Skill Level Assignment Method (SLAM) 1:2 - Continued.

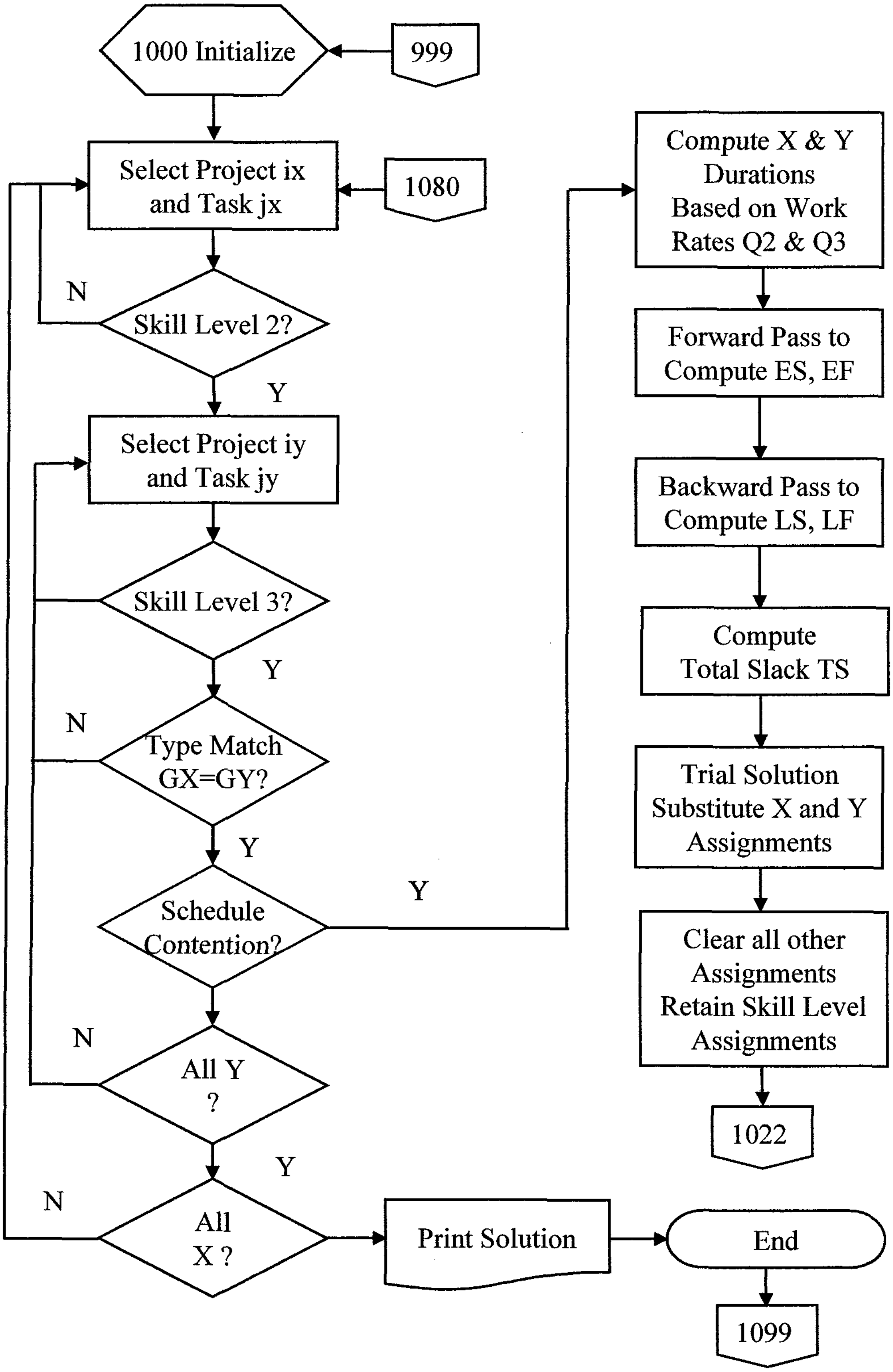

1000 Initialize
999
Select Project ix and Task jx
1080
N
Skill Level 2?
Y
Select Project iy and Task jy
Skill Level 3?
Y
N
Type Match GX=GY?
Y
Schedule Contention?
Y
N
All Y ?
Y
N
All X ?
Print Solution
End
1099
Compute X & Y Durations Based on Work Rates Q2 & Q3
Forward Pass to Compute ES, EF
Backward Pass to Compute LS, LF
Compute Total Slack TS
Trial Solution Substitute X and Y Assignments
Clear all other Assignments Retain Skill Level Assignments
1022

Figure A.14 Flow Chart - Skill Level Assignment Method (SLAM) 2:3.

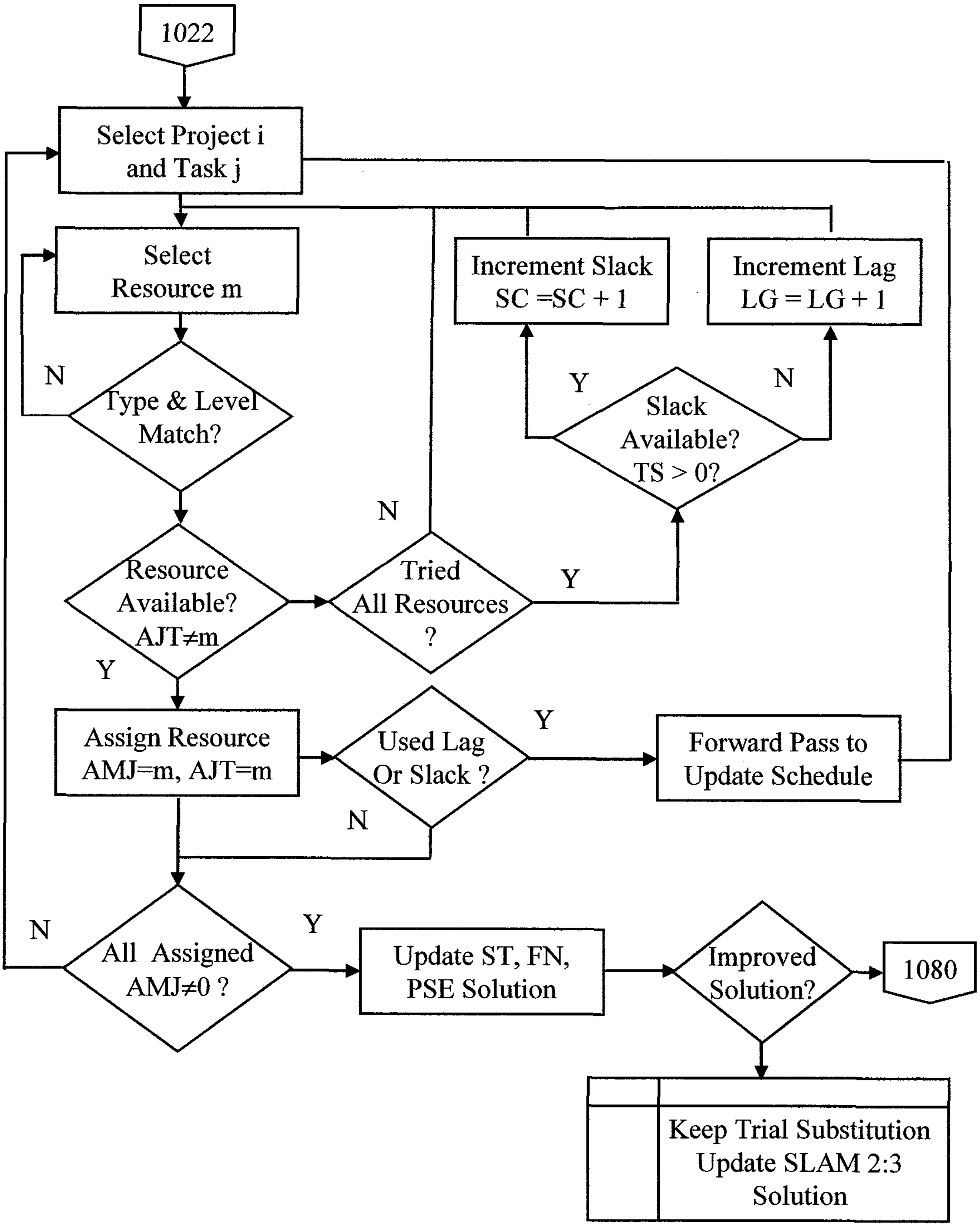
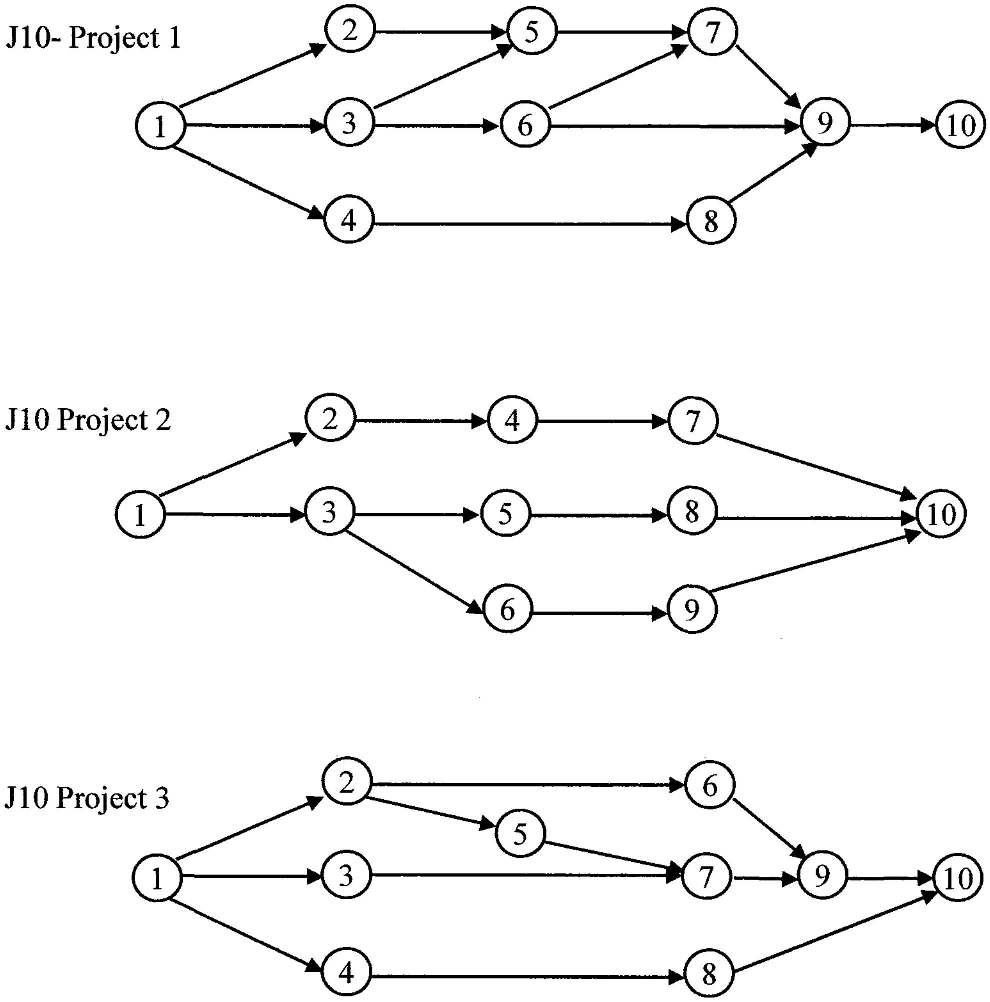

1022
Select Project i and Task j
Select Resource m
N
Type & Level Match?
Increment Slack SC =SC + 1
Increment Lag LG = LG + 1
Y
N
Slack Available? TS > 0?
N
Resource Available? AJT≠m
Tried All Resources ?
Y
Y
Assign Resource AMJ=m, AJT=m
Used Lag Or Slack ?
Y
Forward Pass to Update Schedule
N
N
All Assigned AMJ≠0 ?
Y
Update ST, FN, PSE Solution
Improved Solution?
1080
Keep Trial Substitution Update SLAM 2:3 Solution

Figure A.15 Flow Chart - Skill Level Assignment Method (SLAM) 2:3 - Continued.

APPENDIX B

PROJECT INPUT DATA

Project data sets used in the computational analysis are contained in Appendix B.

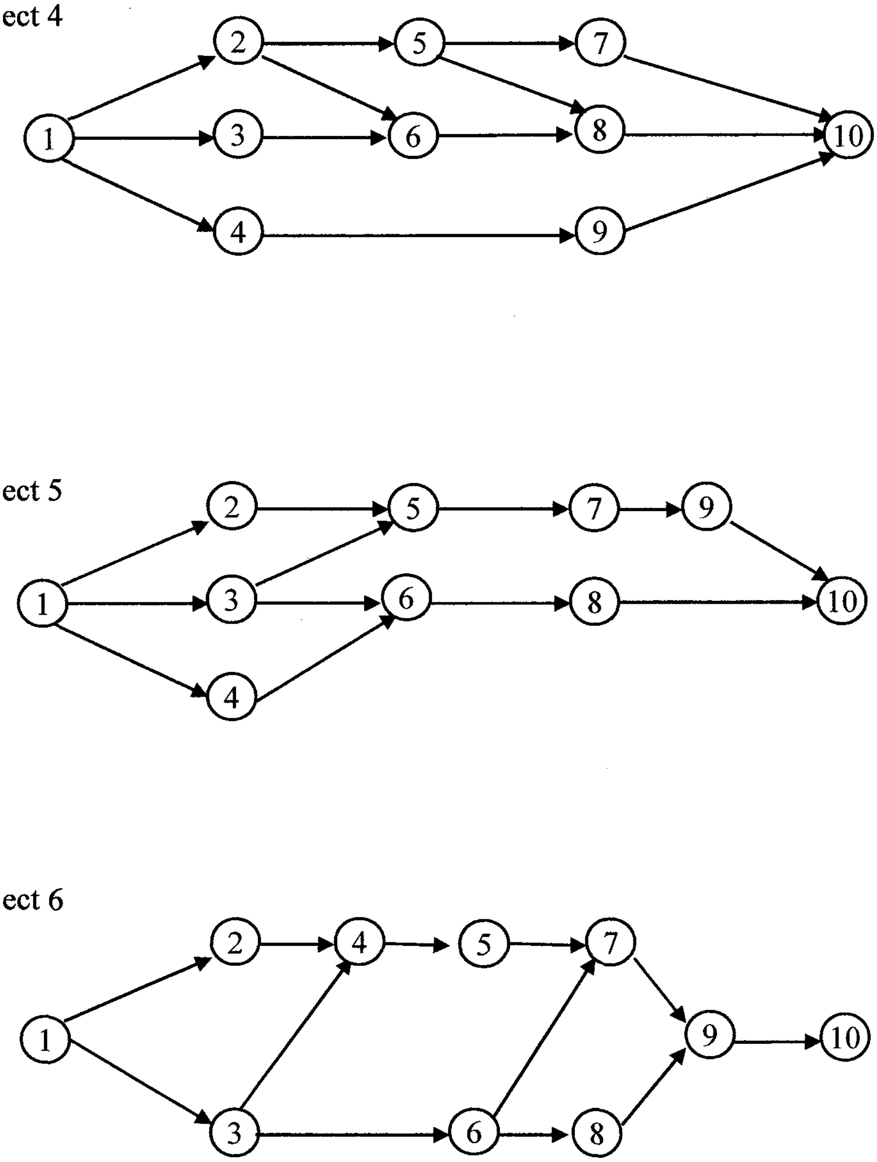


J10- Project 1

J10 Project 2

J10 Project 3

Figure B.1 J10 project networks.



J10 Project 4

J10 Project 5

J10 Project 6

Figure B.2 J10 project networks - continued.

i	1		PROJECT 1							
VP	$17,000,000									
TP	90									
LP	400									
j	1	2	3	4	5	6	7	8	9	10
DM	0	14	28	21	14	28	21	35	21	0
DL	0	21	35	35	28	35	35	49	28	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	1	0	1	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0
i	2		PROJECT 2							
VP	$21,000,000									
TP	90									
LP	800									
j	1	2	3	4	5	6	7	8	9	10
DM	0	21	35	21	42	14	28	28	21	0
DL	0	28	49	28	56	21	42	42	35	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0
5	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

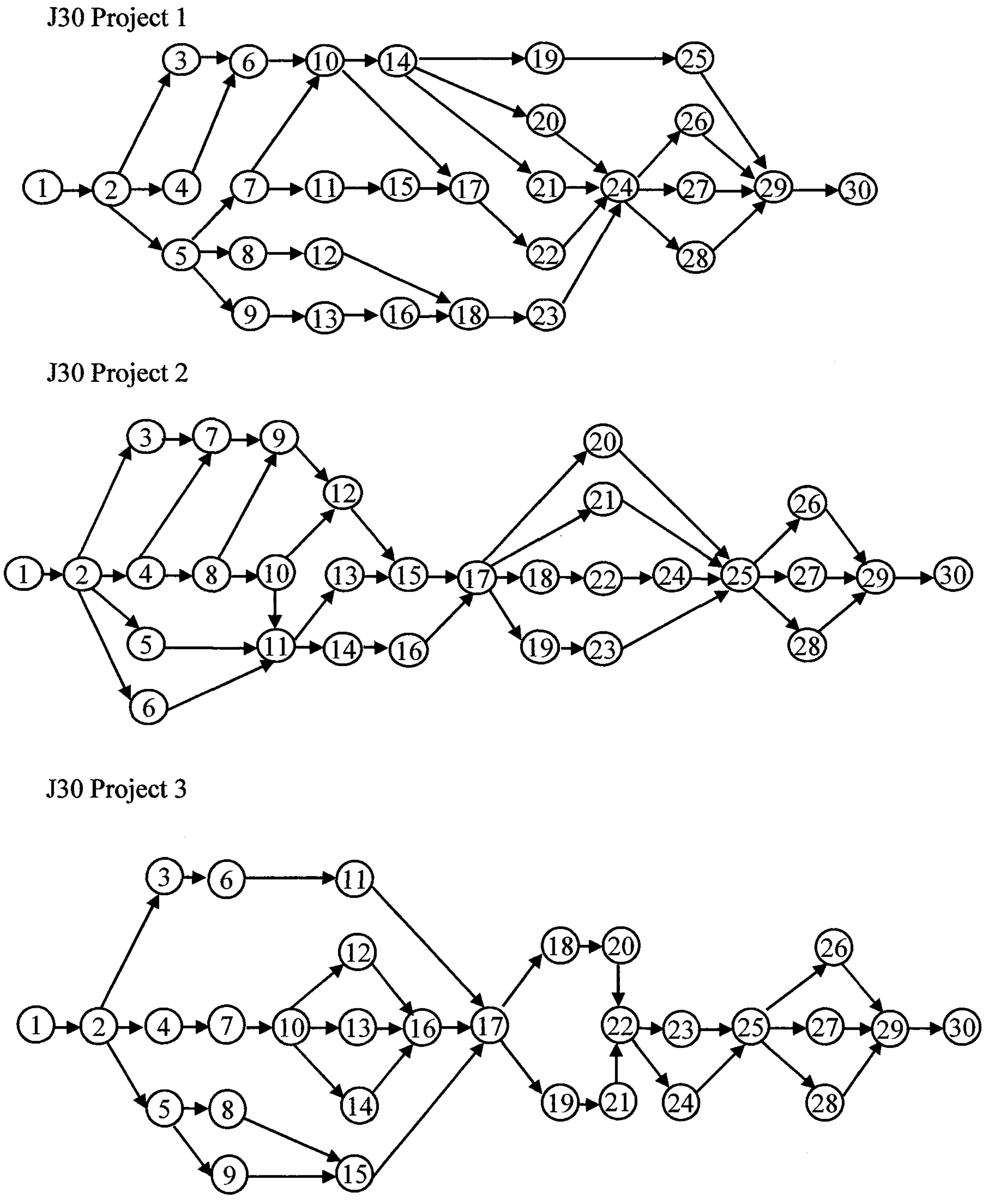
Figure B.3 J10 data input file.

i	3		PROJECT 3							
VP	$22,000,000									
TP	90									
LP	1200									
j	1	2	3	4	5	6	7	8	9	10
DM	0	21	21	21	14	28	35	21	21	0
DL	0	35	35	35	21	42	49	35	35	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0
i	4		PROJECT 4							
VP	$18,000,000									
TP	90									
LP	400									
j	1	2	3	4	5	6	7	8	9	10
DM	0	7	35	14	35	14	35	21	35	0
DL	0	14	42	28	42	28	42	35	42	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	1	1	0	0
6	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Figure B.4 J10 data input file - continued.

i	5		PROJECT 5							
VP	$19,000,000									
TP	90									
LP	800									
j	1	2	3	4	5	6	7	8	9	10
DM	0	21	28	35	21	21	28	21	21	0
DL	0	28	49	42	35	28	49	35	28	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0
i	6		PROJECT 6							
VP	$20,000,000									
TP	90									
LP	1200									
j	1	2	3	4	5	6	7	8	9	10
DM	0	35	28	21	28	7	35	21	7	0
DL	0	42	35	35	49	14	56	35	14	0
FJ	0	1	1	2	2	2	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0
PRC	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	1	1	0	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Figure B.5 J10 data input file - continued.



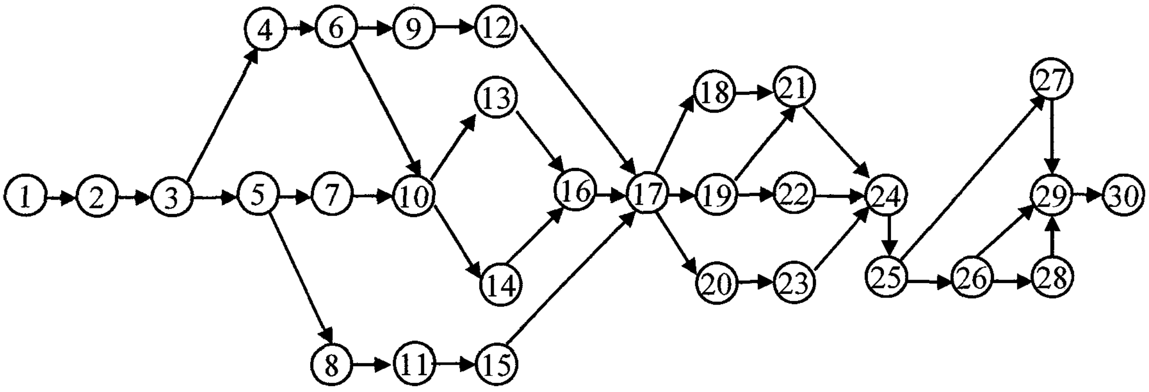
J30 Project 1

J30 Project 2

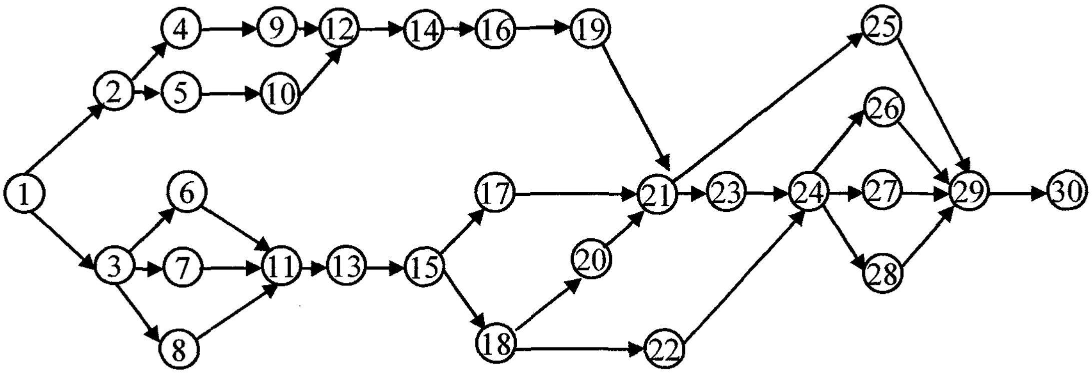
J30 Project 3

Figure B.6 J30 project networks.

J30 Project 4



J30 Project 5



J30 Project 6

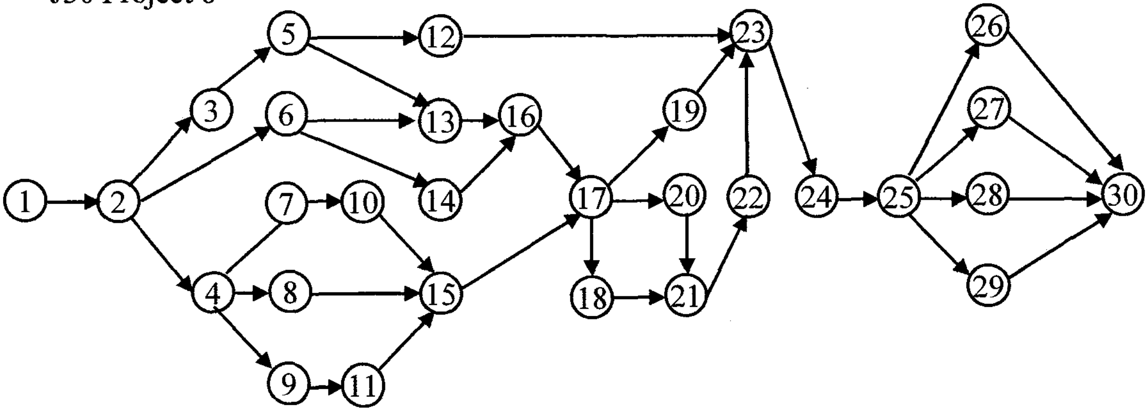


Figure B.7 J30 project networks - continued.

i	1																													
VP	$300,000,000														J30 Project 1															
TP	200																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	35	21	14	28	14	21	35	21	21	21	14	21	28	21	14	14	21	28	35	28	21	14	35	14	21	14	14	35	0
DL	0	49	35	21	35	21	35	56	35	28	35	28	35	35	35	21	21	42	35	49	35	28	21	42	21	28	35	21	49	0
FJ	0	1	1	2	1	1	2	2	4	1	3	2	3	2	3	1	2	4	3	3	3	2	4	2	3	3	1	2	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	3																													
VP	$350,000,000														J30 Project 2															
TP	200																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	28	14	21	28	14	35	21	21	35	14	35	21	21	14	28	35	28	28	35	14	35	35	14	28	14	28	21	28	0
DL	0	35	21	35	35	21	49	35	42	42	21	56	35	35	21	35	42	42	42	56	28	49	42	21	35	28	42	35	35	0
FJ	0	1	1	2	1	3	4	1	3	1	3	3	2	3	4	2	1	2	2	3	3	2	4	2	3	3	1	2	2	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	3																													
VP	$260,000,000														J30 Project 3															
TP	200																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	28	14	21	28	28	14	35	21	35	21	35	21	21	14	28	35	28	21	35	28	35	14	14	28	21	28	14	28	0
DL	0	35	21	35	35	35	28	49	42	42	28	49	35	35	21	35	42	42	35	56	42	49	21	21	35	35	42	28	35	0
FJ	0	1	1	2	3	4	1	2	1	3	4	3	1	2	3	2	1	2	3	3	2	2	4	3	1	2	3	4	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	4																													
VP	$320,000,000														J30 Project 4															
TP	300																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	14	28	21	28	14	28	35	21	14	28	35	21	21	14	28	21	28	28	21	14	35	21	14	28	21	28	14	28	0
DL	0	21	35	35	35	28	35	56	42	21	49	56	35	35	21	35	42	42	42	35	21	49	35	21	35	35	42	21	35	0
FJ	0	1	1	1	2	1	3	2	1	4	3	3	1	2	3	2	1	2	3	4	2	2	3	3	1	2	3	4	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	5																													
VP	$220,000,000														J30 Project 5															
TP	300																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	14	28	21	28	21	14	35	21	21	28	35	21	21	14	28	28	21	14	35	28	35	21	14	28	14	28	21	28	0
DL	0	28	35	35	35	35	28	56	42	42	49	49	35	35	21	35	42	35	28	56	42	49	42	21	35	28	42	35	35	0
FJ	0	1	1	3	2	1	2	2	4	3	3	2	3	2	1	4	2	3	3	2	2	2	4	3	1	2	3	2	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6																													
VP	$280,000,000														J30 Project 6															
TP	300																													
LP	1000																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	14	28	21	28	14	28	35	21	35	21	35	14	21	14	28	35	28	21	35	21	35	28	14	28	21	28	14	28	0
DL	0	28	35	35	35	21	35	56	42	42	35	56	28	35	21	35	42	42	35	56	35	49	42	21	35	35	42	28	35	0
FJ	0	1	3	1	2	1	4	2	3	3	2	2	4	3	2	1	2	2	3	3	2	2	4	3	3	1	2	3	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure B.8 J30 data input file – shown without Prec matrix.

J60 Project 1

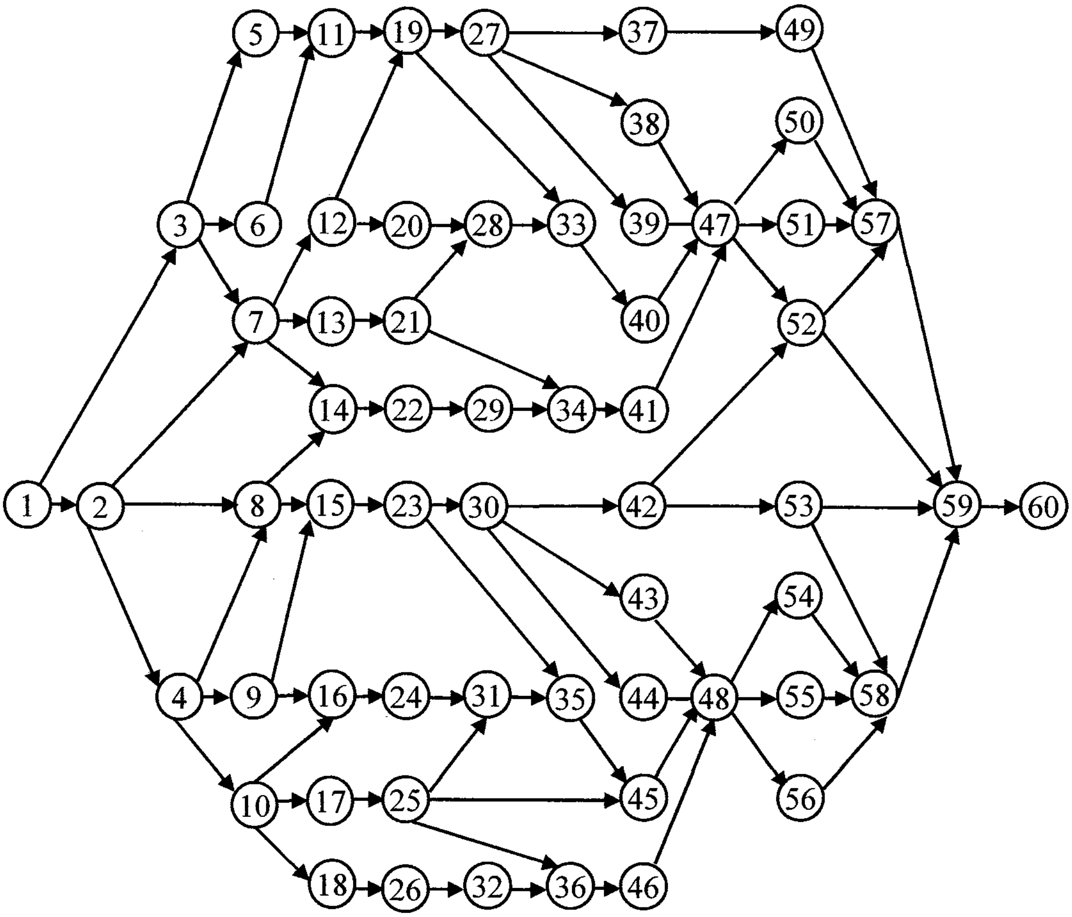


Figure B.9 J60 project networks.

J60 Project 2

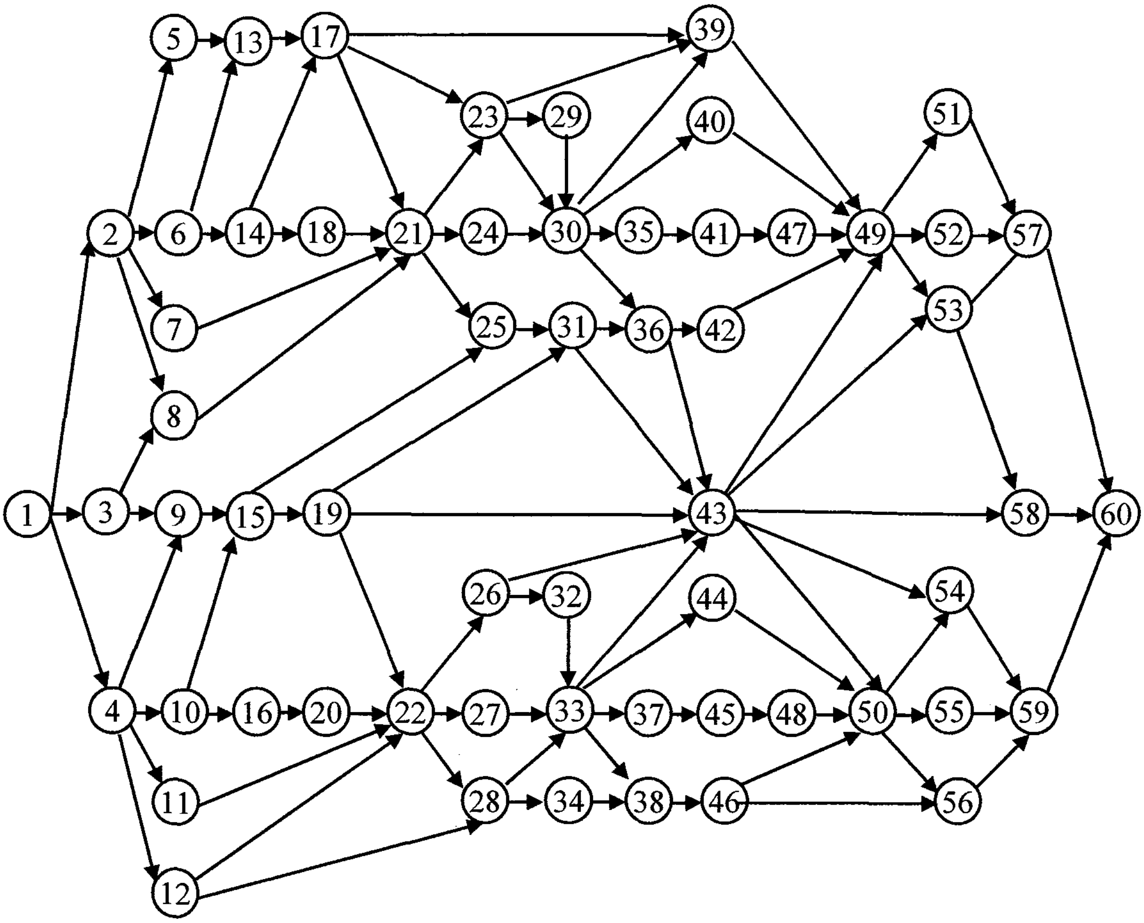


Figure B10 J60 project networks - continued.

J60 Project 3

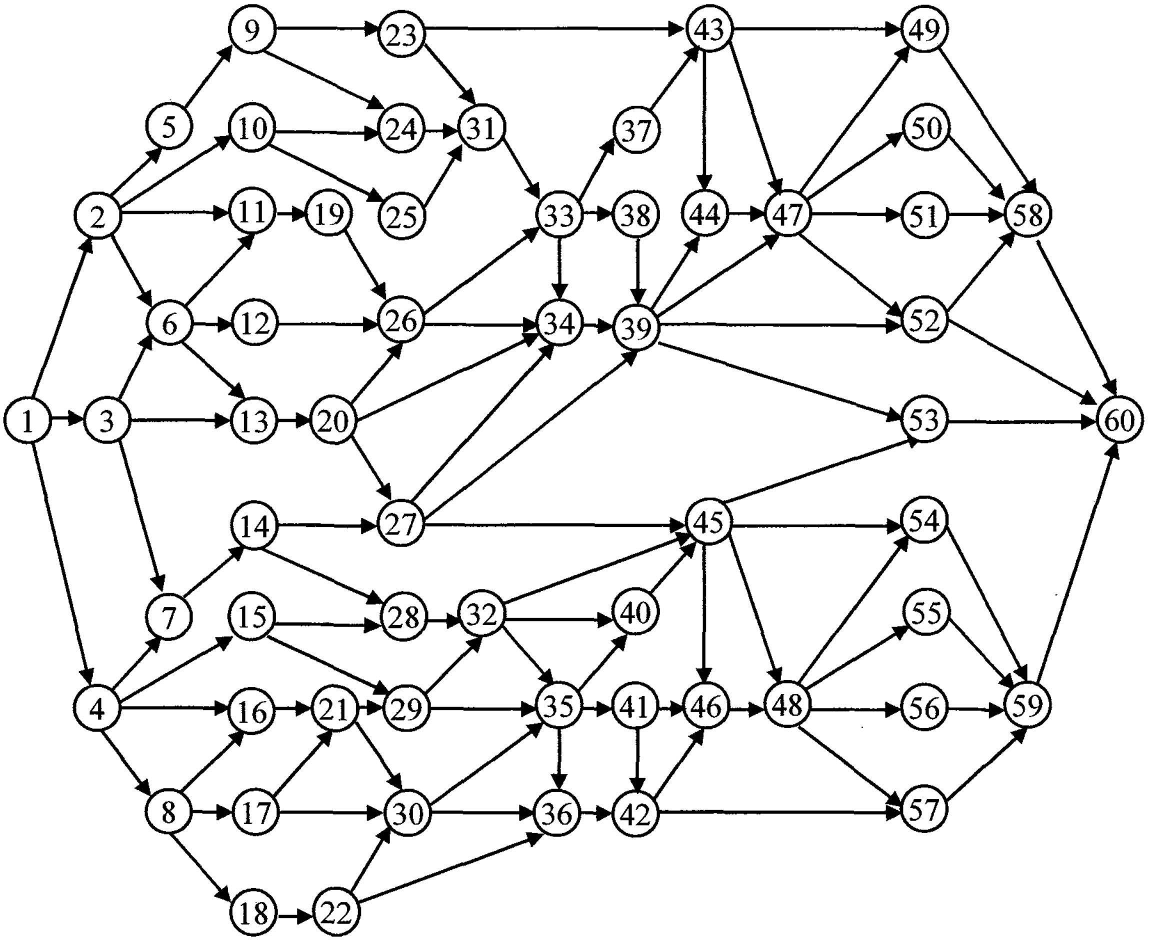


Figure B11 J60 project networks - continued.

J60 Project 4

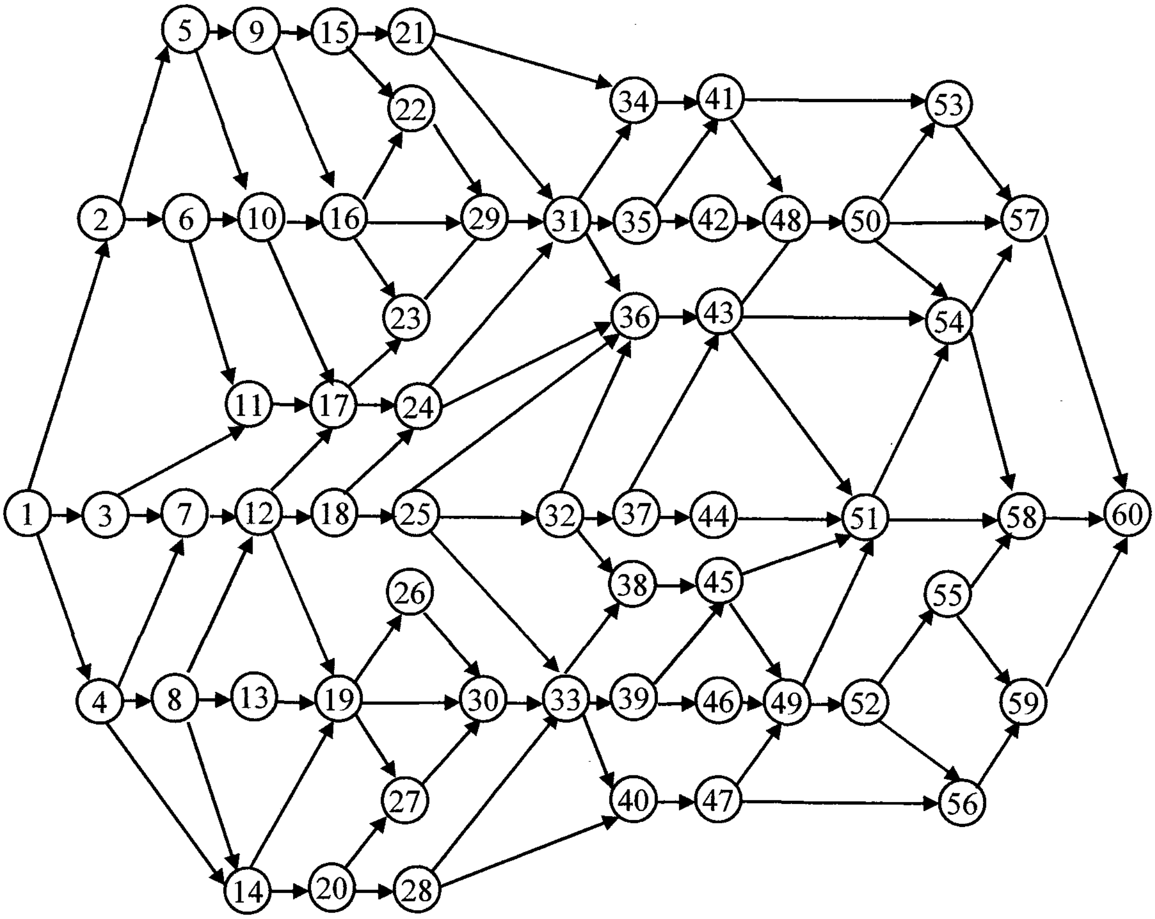


Figure B12 J60 project networks – c ontinued.

i	1																													
VP	$350,000,000																J60 Project 1													
TP	300																													
LP	720																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	7	21	14	7	14	7	14	21	28	21	35	35	28	21	14	14	21	28	7	14	7	14	35	14	28	14	14	21	7
DL	0	14	35	21	21	21	14	21	35	35	35	49	77	35	35	21	21	42	35	14	21	21	21	42	21	35	35	21	28	14
FJ	0	1	1	2	4	1	2	3	2	1	5	2	3	4	3	1	4	2	3	5	3	2	5	2	3	3	1	4	3	2
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Task	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
DM	7	21	7	14	14	14	21	35	21	14	21	21	7	14	21	14	14	21	7	28	14	28	14	7	14	28	14	7	28	0
DL	14	35	14	21	21	21	28	56	35	21	35	28	21	21	35	21	21	42	14	42	28	35	21	21	21	35	35	21	35	0
FJ	3	1	1	2	4	1	2	4	2	1	3	2	3	4	3	1	2	5	3	3	3	2	5	2	3	4	1	2	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	1																													
VP	$450,000,000																J60 Project 2													
TP	300																													
LP	720																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	7	21	14	14	14	21	14	21	28	21	35	35	28	21	14	14	21	28	35	7	42	14	28	14	28	14	21	14	7
DL	0	14	35	21	21	21	28	28	35	98	35	49	77	42	35	21	21	42	35	49	14	56	21	42	21	35	35	42	28	14
FJ	0	1	1	2	1	4	2	2	5	1	3	2	3	5	3	1	2	4	3	3	3	2	4	2	3	3	1	2	3	4
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
DL	7	21	21	14	14	14	14	7	21	28	21	35	7	28	21	14	14	21	28	28	21	21	14	28	14	28	14	28	7	0
FJ	14	35	35	21	21	21	28	21	35	98	35	49	21	42	35	21	21	42	35	42	35	28	21	42	21	35	35	42	21	0
LG	2	1	5	2	1	1	2	4	2	1	3	2	3	4	3	1	2	2	3	3	5	2	2	2	3	4	1	2	3	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	1																													
VP	$400,000,000																J60 Project 3													
TP	300																													
LP	720																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	21	7	14	14	14	21	7	21	28	21	28	14	21	21	14	14	21	28	7	14	21	14	35	14	28	14	28	14	7
DL	0	35	21	21	21	21	28	28	35	42	35	42	28	28	35	21	21	42	35	14	21	35	21	42	21	35	35	42	21	14
FJ	0	1	1	2	1	1	2	4	2	1	3	2	4	2	3	1	2	4	3	5	3	2	2	5	3	3	1	2	3	3
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Task	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
DM	7	21	21	14	14	14	21	14	21	21	21	35	14	21	21	14	14	21	28	14	28	21	14	21	14	28	14	28	12	0
DL	14	35	35	21	21	21	35	28	35	28	35	49	28	35	35	21	21	42	35	21	42	35	21	42	21	35	35	42	35	0
FJ	5	1	1	2	1	1	2	4	2	1	3	2	3	2	3	1	2	2	5	3	3	2	2	4	3	3	1	2	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	1																													
VP	$500,000,000																J60 Project 4													
TP	365																													
LP	720																													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
DM	0	21	7	14	14	14	42	35	21	28	21	35	35	21	21	14	14	21	28	7	35	42	14	21	14	28	14	28	21	7
DL	0	35	14	21	21	21	56	56	35	98	35	49	77	28	35	21	21	42	35	14	56	56	21	42	21	35	35	42	35	14
FJ	0	1	1	2	1	1	2	2	2	1	3	2	3	2	5	1	2	2	3	3	5	2	2	4	3	3	1	2	3	2
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Task	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
DM	7	21	21	14	14	14	21	35	21	28	21	21	21	56	21	14	14	21	7	35	35	21	14	35	14	28	14	7	28	0
DL	14	35	35	21	21	21	28	49	35	42	35	28	35	77	35	21	21	42	21	49	56	35	21	42	21	35	35	21	35	0
FJ	4	1	1	2	1	4	2	2	5	1	3	2	3	2	3	1	5	2	3	4	3	2	2	2	4	3	1	2	3	0
LG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure B.13 J60 project data file – shown without Prec matrix.

APPENDIX C

CODE LISTING

The source code for the Resourch Assignment and Multi-Project Scheduling (RAMPS) program is found in this appendix. Flow charts for each sub-procedure are found in Appendix A. The scheduling and assignment methods were programmed in Visual Basic for Applications (VBA) contained within Microsoft Excel. The VBA platform allows convenient input and output from Excel worksheets and allows the entire analysis to be contained within a workbook. Sub-procedures are provided with line number references corresponding to the procedure; for example line numbers 100 to 199 contains code for the Robust Scheduling Method. The high level program structure is as follows:

Declarations

Main Procedure

Sub-Procedures:

010	Initialization
100	Robust Scheduling Method
200	Project Valuation Method
300	Project Prioritization Method
400	Resource Prioritization Method
500	Sequential Scheduling Method
600	Critical Scheduling Method
700	Early Scheduling Method
800	Skill Level Scheduling Method
900	Skill Level 1:2 Assignment Method
1000	Skill level 2:3 Assignment Method

Print Results

Print Charts

The program code listed in this appendix provides solutions for the "J10" data set consisting of six projects each having ten tasks. The code arrays are dimensioned for six projects (NP=6), ten tasks (JT=10), three resource types (KT=3), up to sixty resources (MR=60) and up to eight-hundred (TP=800) time periods.

This program was expanded to conduct computation analysis the larger data sets including "J30" and "J60". The expanded code (not shown) was created by modifying arrays in the code to reflect the values shown in Table C.1.

Table C.1 –Array values for project data sets.

Data Set	NP Number of Projects	JT Number of Tasks	KT Number of Resource Types	MR Number of Resources	TP Number of Time Periods (days)
J10	6	10	3	60	800
J30	6	30	4	120	1000
J50	4	60	5	180	1000

```
Option Base 1

'Global Declarations

'Resources
Dim MAX As Integer                  'Maximum number of resources
Dim MR As Integer                   'Number of resources
Dim MRMAX As Integer               'Number of Resources Derived by the RPM
Dim MMR As Integer                 'Set MMR=1 with multiple resource sets
Dim KT As Integer                  'Number of resource types, e.g. 3
Dim LT As Integer                   'Number of skill levels, e.g. 3
Dim m As Integer                   'Index of resources
Dim GM(60, 60) As Integer           'Skill type of resource m, GM = 1, 2 or 3
Dim HM(60, 60) As Integer           'Skill level of resource m, HM = 1,2 or 3
Dim LRI(60, 60) As Integer          'Resource Index prioritized by Skill Level

Dim RRI(60) As Integer              'Random Resource Index
Dim RRA(60) As Single               'Random Variable for prioritizing resources
Dim RRB(60) As Single               'Random Variable for prioritizing resources
Dim RPRB(60) As Integer             'Priority resource random variable B
Dim Q1 As Variant                   'Work rate level 1 increased over level 2
Dim Q3 As Variant                   'Work rate level 3 reduced from level 2


'Schedule Time Periods
Dim TP As Integer                  'Number of time periods
Dim AJT(6, 10, 800) As Integer     'Assignment resource m to task j in period T
Dim AJTO(6, 10, 800) As Integer    'Assignment Printing variable
Dim BAJT(6, 10, 800) As Integer     'Assignment matrix for best solution

'Prioritization Parmeters
Dim PP As Integer                   'PP =0 Sequential order per input file
                                    'PP=1 Productivity Prioritized
                                    'PP=2 NPVP Prioritized, PP=3 Random

Dim PR As Integer                  'Prioritization of Resources
                                    'PR=0 Un-prioritized, as generated or input
                                    'PR=1 Prioritized by Skill Level, highest first

Dim Alpha As Variant                'Likelyhood of recovering lost value
Dim Beta As Variant                'Likelyhood of additional loss
Dim IR As Variant                   'Interest Rate per day
Dim Delta As Variant                'Near critical tasks, percent of CP

Dim Kappa1 As Variant              'Workforce parameter, % Resource Type 1(0 to 1)
Dim Kappa2 As Variant               'Workforce parameter, % Resource Type 2(0 to 1)
Dim Kappa3 As Variant           'Workforce parameter, % Resource Type 3(0 to 1)

Dim Lambda1 As Variant          'Workforce parameter, % Skill Level 1(0 to 1)
Dim Lambda2 As Variant          'Workforce parameter, % Skill Level 2(0 to 1)
Dim Lambda3 As Variant          'Workforce parameter, % Skill Level 3(0 to 1)

'Declarations
Dim NP As Integer               'Number of projects
Dim NPC As Integer              'Number of critical projects
Dim JT As Integer               'Number of tasks
Dim NJT As Integer              'Total Number of tasks, NP*JT
Dim STEP As Variant             'Incremental value in ES sort

Dim i As Integer                'Index of projects
Dim j As Integer                'Index of tasks
Dim k As Integer                'Index of skill types
Dim l As Integer                'Index of skill levels

Dim n As Integer                'Project counter
Dim p As Integer                'Task counter
Dim t As Integer                'Index of time periods
Dim TT As Integer               'Index of time periods
Dim ii As Integer               'Index of projects
Dim jj As Integer               'Index of revised tasks
Dim iii As Integer              'Index of projects
Dim jjj As Integer              'Index of tasks
Dim x As Integer                'Row reference in Project Input
Dim y As Integer                'Col reference in Project Input
Dim R As Integer                'Row reference in UPS
Dim C As Integer                'Col reference in UPS
Dim jr As Integer               'Row reference for precedence matrix
Dim jc As Integer               'Col reference for precedence matrix
Dim Row As Integer              'Row reference for Assignment Output
Dim Col As Integer              'Col reference for Assignment Output
Dim Excelrow As Integer         'Row reference for output
Dim Excelcol As Integer         'Col reference for output
Dim Resultsrow As Integer       'Row reference for results printing
Dim testrow As Integer          'Row reference for diagnostic output

Dim TRIAL12 As Integer          'No. of Trial Level 1:2 Substitutions
Dim TRIAL23 As Integer          'No. of Trial Level 2:3 Substitutions
Dim SUB12 As Integer            'No of Level 1:2 Substitutions
Dim SUB23 As Integer            'No of Level 2:3 Substitutions

Dim PI(6) As Integer            'Project Number
Dim MAXEFPRC(6) As Integer      'Maximum Early Finish of Predecessors
Dim MINLSSUC(6) As Integer      'Minimum Late Start of Successors
Dim MINESSUC(6) As Integer      'Minimum Early Start of Successors
Dim MAXTEFPRC(6) As Integer     'Maximum Temp Early Finish of Predecessors

Dim RS(6) As Integer            'Indicator to reschedule project
Dim CP(6) As Integer            'Critical Path Duration for each project
Dim SS(6) As Variant            'Square root of Sum of Square Errors each project
Dim BB(6) As Variant            'Bias Buffer Estimate
Dim BF(6) As Variant            'Total Buffer Estimate Includes SS and BB
Dim CC(6) As Integer            'Critical Chain Duration for each project
Dim SV(6) As Integer            'Sum value
Dim VP(6) As Variant            'Potential Value of the Project
Dim VE(6) As Variant            'Expected Value of the Project based on TE
Dim VL(6) As Variant            'Lost Value to the Project; VP-VE
Dim LP(6) As Integer            'Potential Life - days
Dim TL(6) As Integer            'Potential Time to Launch -days
Dim TE(6) As Integer            'Expected project duration
Dim DE(6) As Integer            'Expected Delay
Dim NPVP(6) As Variant          'Potential Net Present Value
Dim NPVE(6) As Variant          'Expected Net Present Value
Dim NPVL(6) As Variant          'Lost Net Present Value due to delay
Dim GL(6) As Variant            'Left hand gradiant for potential and expected
Dim GR(6) As Variant            'Right hand gradiant for expected
Dim GAMMA(6) As Variant         'Rate of loss factor
Dim PPM(6) As Variant           'Project Productivity Metric
Dim PRI(6) As Integer           'Productivity Ranking Index
Dim RPRI(6) As Integer          'Random Priority Ranking Index
Dim PRB(6) As Integer           'Temporary Random Priority Index
Dim RA(6) As Single             'Random Number
Dim RB(6) As Single             'Temporary Array Random Number
Dim RTEMP    As Variant         'Random Tempory Value
Dim ESA(60) As Variant          'Early Start sorting variable
Dim ESB(60) As Variant          'Early Start sorting variable
Dim ESARO(60) As Variant        'Early Start in rank order
Dim PESA(60) As Integer         'Early Start Priority
Dim PESB(60) As Integer         'Early Start Priority

Dim SSMTCS(60) As Variant       'Sequential Time Constrained Solution
Dim SSMRCS(60) As Variant       'Sequential Resource Constrained Solution

Dim CSMTCS(60) As Variant       'Critical Time Constrained Solution
Dim CSMCCS(60) As Variant       'Critical Critical Constrained Solution
Dim CSMRCS(60) As Variant       'Critical Resource Constrained Solution

Dim ESMTCS(60) As Variant       'Early Time Constrained Solution
Dim ESMRCS(60) As Variant       'Early Resource Constrained Solution
Dim LCS(60) As Variant              'Leveled Solution after applying work rates
Dim SLAM12(60) As Variant           'Skill Level Assignment Method Levels 1 and 2
Dim SLAM23(60) As Variant           'Skill Level Assignment Method Levels 2 and 3

Dim RC(60) As Variant               'Resource Cost, for each set
Dim RC1 As Variant                  'Cost/day of Level 1 Resource
Dim RC2 As Variant                  'Cost/day of Level 2 Resource
Dim RC3 As Variant                  'Cost/day of Level 3 Resource
Dim CF(60) As Variant               'Fractional Cost

Dim PTEMP(6) As Variant             'Used in Project Ranking/Sorting
Dim TEMP As Variant                 'Used in Project Ranking/Sorting
Dim VPTEMP(6) As Variant            'Used in Project Ranking
Dim VPTEMPPRI(6) As Variant         'Used in Project Ranking
Dim VPRI(6) As Variant              'Potential Value-Revenue Priority Ranking Index"
Dim PRITEMP(6) As Integer           'Prioritization Variable
Dim NPVPTEMP(6) As Variant          'Potential Net Present Value
Dim NPVPPRI(6) As Variant           'Priority based on Potential Net Present Value
Dim NPVPTEMPPRI(6) As Variant       'Temp prioritiy based on Potential Net Present
Value
Dim NPTEMP As Variant               'Net Present Value
Dim NPVETEMP(6) As Variant          'Expected Net Present Value
Dim TEMPNPVE(6) As Variant          'Expected Net Present Value
Dim NPVEPRI(6) As Variant           'Priority Based on Expected Net Present Value
Dim NPVETEMPPRI(6) As Variant       'Priority Based on Expected Net Present Value
Dim NETEMP As Variant               'Temporary Value
Dim SUMNPVP As Variant              'Sum of Program Potential Net Present Value
Dim SUMNPVE As Variant              'Sum of Program Expected Net Present Value
Dim PSE As Variant                  'Program Scheduling Efficiency

Dim GPL(6) As Variant               'Gradiant for Potential Triangle Left
Dim GPR(6) As Variant               'Gradiant for Potential Triangle Right
Dim GEL(6) As Variant               'Gradiant for Expected Triangle Left
Dim GER(6) As Variant               'Gradiant for Expected Triangle Right
Dim RPK(6) As Variant               'Peak Value of Potential Triangle
Dim REK(6) As Variant               'Peak Value of Expected Triangle
Dim DFPGLP(6) As Variant            'Discount Factor P/G Gradiant left potential
Dim DFPFLP(6) As Variant            'Discount Factor P/F for left potential triangle
Dim DFPARP(6) As Variant            'Discount Factor P/A for RP potential annuity
Dim DFPGRP(6) As Variant            'Discount Factor P/G Gradiant for right potential
Dim DFPFRP(6) As Variant            'Discount Factor P/F for right potential triangle
Dim DFPGLE(6) As Variant            'Discount Factor P/G Gradiant left expected
Dim DFPFLE(6) As Variant            'Discount Factor P/F left expected triangle
Dim DFPARE(6) As Variant            'Discount Factor P/A RE expected annuity
Dim DFPGRE(6) As Variant            'Discount Factor P/G Gradiant right expected
Dim DFPFRE(6) As Variant              'Discount Factor P/F for right expected triangle
Dim DV(6) As Variant                  'Derivative of the value loss function
Dim MAXDV As Variant                  'Maximum value of DV
Dim MAXNPVE As Variant                'Maximum value of NPVE set as hurdle for
improvement

Dim DM(6, 10) As Integer              'Mean task duration estimate
Dim DL(6, 10) As Integer              'Low risk task duration estimate
Dim TDM(6, 10) As Integer             'Temporary mean task duration estimates
Dim TDL(6, 10) As Integer             'Temporary low risk task duration estimates
Dim FJ(6, 10) As Integer              'Required skill type for task j
Dim ES(6, 10) As Integer              'Earliest Start Time task j
Dim EF(6, 10) As Integer              'Earliest Finish Time task j
Dim LS(6, 10) As Integer              'Late Start Time task j
Dim LF(6, 10) As Integer              'Late Finish Time task j

Dim LG(6, 10) As Integer              'Lag Time for starting task j
Dim LC(6, 10) As Integer              'Lag Counter
Dim TS(6, 10) As Integer              'Total slack in unconstrained model
Dim WS(6, 10) As Integer              'Working slack array
Dim SC(6, 10) As Integer              'Slack counter
Dim FS(6, 10) As Integer              'Free slack task j
Dim DS(6, 10) As Integer              'Decrease in slack
Dim SE(6, 10) As Integer              'Square error each task j
Dim CT(6, 10) As Integer              'Critical Path Task Indicator 1/0
Dim NC(6, 10) As Integer              'Near Critical Path Task Indicator 1/0
Dim AMJ(6, 10) As Integer             'Assignment of resource m to task j
Dim AHJ(6, 10) As Integer             'Assignment of level to a particular task
Dim ST(6, 10) As Integer              'Start Time for task j
Dim FN(6, 10) As Integer              'Finish Time for task j
Dim MA(6, 10) As Integer              'Resource M Availability Indicator 1/0
Dim PRC(6, 10, 10) As Integer         'Precedence Matrix

'Print Variables
Dim CPO(6) As Integer                 'Critical Path
Dim BFO(6) As Variant                 'Project Buffer
Dim TEO(6) As Integer                 'Expected Duration
Dim DEO(6) As Integer                 'Expected Delay Output
Dim NPVEO(6) As Variant               'Expected Net Present Value
Dim DMO(6, 10) As Integer             'Mean Duration Estimate
Dim DLO(6, 10) As Integer             'Low Risk Duration Estimate
Dim FJO(6, 10) As Integer             'Task Resource Type Requirement
Dim ESO(6, 10) As Integer             'Early Start Date
Dim EFO(6, 10) As Integer             'Early Finish Date
Dim LSO(6, 10) As Integer             'Late Start Date
Dim LFO(6, 10) As Integer             'Late Finish Date
Dim STO(6, 10) As Integer          'Start Date
Dim FNO(6, 10) As Integer          'Finish Date
Dim CTO(6, 10) As Integer          'Critical Task Indicator
Dim TSO(6, 10) As Integer          'Total Slack
Dim FSO(6, 10) As Integer          'Free Slack
Dim WSO(6, 10) As Integer          'Working Slack
Dim SCO(6, 10) As Integer          'Slack Counter
Dim LGO(6, 10) As Integer          'Lag
Dim AMJO(6, 10) As Integer         'Assignment Matrix, resource m to task j
Dim AHJO(6, 10) As Integer         'Assignment Matrix, level h to task j
Dim PSEO As Variant                'Program Schedule Efficiency
Dim SUMNPVEO As Variant            'Summation Expected Net Present Value
Dim SUMNPVPO As Variant            'Summation Potential Net Present Value

'Best RCS Solution
Dim BEST(60) As Integer            'Best Resource Constrained Solution
Dim BRCS(60) As Variant            'Best Resource Constrained Solution
Dim BCP(6) As Integer              'Critical Path
Dim BBF(6) As Variant              'Project Buffer
Dim BTE(6) As Integer              'Expected Project Duration
Dim BDE(6) As Integer              'Expected Delay
Dim BNPVE(6) As Variant            'Expected Net Present Value
Dim BDM(6, 10) As Integer          'Mean Duration Estimate
Dim BDL(6, 10) As Integer          'Low Risk Duration Estimate
Dim BFJ(6, 10) As Integer          'Task Resource Type Requirement
Dim BES(6, 10) As Integer          'Early Start Date
Dim BEF(6, 10) As Integer          'Early Finish Date
Dim BLS(6, 10) As Integer          'Late Start Date
Dim BLF(6, 10) As Integer          'Late Finish Date
Dim BST(6, 10) As Integer          'Start Date
Dim BFN(6, 10) As Integer          'Finish Date
Dim BCT(6, 10) As Integer          'Critical Task Indicator
Dim BTS(6, 10) As Integer          'Total Slack
Dim BFS(6, 10) As Integer          'Free Slack
Dim BWS(6, 10) As Integer          'Working Slack
Dim BSC(6, 10) As Integer          'Slack Counter
Dim BLG(6, 10) As Integer          'Lag
Dim BAMJ(6, 10) As Integer         'Assignment Matrix Resource m to Task j
Dim BAHJ(6, 10) As Integer         'Assignment Matrix Skill Level h to Task j

'Skill-Level Improved Solution
Dim SPSE As Variant                'Program Scheduling Efficiency
Dim SCP(6) As Integer              'Critical Path
Dim SBF(6) As Variant              'Project Buffer
Dim SBB(6) As Variant              'Bias Buffer
Dim SSS(6) As Variant              'Sum of Square Errors
Dim STE(6) As Integer               'Expected Project Duration
Dim SDE(6) As Integer               'Expected Delay
Dim SNPVE(6) As Variant             'Expected Net Present Value
Dim SVE(6) As Variant               'Expected Value
Dim SDM(6, 10) As Integer           'Mean Duration Estimates
Dim SDL(6, 10) As Integer           'Low Risk Duration Estimates
Dim SES(6, 10) As Integer           'Early Start Date
Dim SEF(6, 10) As Integer           'Early Finish Date
Dim SLS(6, 10) As Integer           'Late Start Date
Dim SLF(6, 10) As Integer           'Late Finish Date
Dim SST(6, 10) As Integer           'Start Date
Dim SFN(6, 10) As Integer           'Finish Date
Dim SCT(6, 10) As Integer           'Critical Task Indicator
Dim STS(6, 10) As Integer           'Total Slack
Dim SFS(6, 10) As Integer           'Free Slack
Dim SWS(6, 10) As Integer           'Working Slack
Dim SSC(6, 10) As Integer           'Slack Counter
Dim SLG(6, 10) As Integer           'Lag
Dim SSE(6, 10) As Variant           'Sum Square Error
Dim SAMJ(6, 10) As Integer          'Assignment Matrix Resource m to Task j
Dim SAHJ(6, 10) As Integer          'Assignment Matrix Skill Level h to Task j
Dim SAJT(6, 10, 800) As Integer     'Assignment Resource m to Task j in Period t

'SLAM 1:2 Improved solution
Dim UPSE As Variant                 'Program Scheduling Efficiency
Dim UCP(6) As Integer               'Critical Path
Dim UBB(6) As Single                'Bias Buffer
Dim UBF(6) As Single                'Total Buffer
Dim UCC(6) As Integer               'Critical Chain Duration
Dim UTE(6) As Integer               'Expected project duration, equivalent to CCD
Dim UDE(6) As Integer               'Expected Delay potential vs expected duration
Dim USS(6) As Integer               'Sum of Square Errors
Dim UVE(6) As Variant               'Expected Value
Dim UNPVE(6) As Single              'Expected Net Present Value
Dim USUMNPVE As Single              'Sum of NPVE
Dim UDM(6, 10) As Integer           'Mean task duration estimates
Dim UDL(6, 10) As Integer           'Low risk task duration estimates
Dim UES(6, 10) As Integer           'Earliest Start Time task j
Dim UEF(6, 10) As Integer           'Earliest Finish Time task j
Dim ULS(6, 10) As Integer           'Late Start Time task j
Dim ULF(6, 10) As Integer           'Late Finish Time task j
Dim UTS(6, 10) As Integer           'Total slack
Dim UWS(6, 10) As Integer           'Working slack
Dim UFS(6, 10) As Integer           'Free slack
Dim USC(6, 10) As Integer           'Slack counter
Dim ULG(6, 10) As Integer           'Lag
Dim UCT(6, 10) As Integer           'Critical Path Task Indicator 1/0
Dim USE(6, 10) As Integer           'Square error
Dim UST(6, 10) As Integer           'Start Time for task j
Dim UFN(6, 10) As Integer           'Finish Time for task j
Dim UAMJ(6, 10) As Integer          'Resource assignment matrix
Dim UAHJ(6, 10) As Integer          'Skill level assignment matrix
Dim UAJT(6, 10, 800) As Integer     'Assignment resource m to task j in period t


'SLAM 1:2:3 Improved solution
Dim VPSE As Variant                 'Program Scheduling Efficiency
Dim VCP(6) As Integer               'Critical Path
Dim VBB(6) As Single                'Bias Buffer
Dim VBF(6) As Single                'Total Buffer
Dim VCC(6) As Integer               'Critical Chain Duration
Dim VTE(6) As Integer               'Expected project duration, equivalent to CCD
Dim VDE(6) As Integer               'Expected Delay potential vs expected duration
Dim VSS(6) As Integer               'Sum of Square Errors
Dim VVE(6) As Variant               'Expected Value
Dim VNPVE(6) As Single              'Expected Net Present Value
Dim VSUMNPVE As Single              'Sum of NPVE
Dim VDM(6, 10) As Integer           'Mean task duration estimates
Dim VDL(6, 10) As Integer           'Low risk task duration estimates
Dim VES(6, 10) As Integer           'Earliest Start Time task j
Dim VEF(6, 10) As Integer           'Earliest Finish Time task j
Dim VLS(6, 10) As Integer           'Late Start Time task j
Dim VLF(6, 10) As Integer           'Late Finish Time task j
Dim VTS(6, 10) As Integer           'Total slack
Dim VWS(6, 10) As Integer           'Working slack
Dim VFS(6, 10) As Integer           'Free slack
Dim VSC(6, 10) As Integer           'Slack counter
Dim VLG(6, 10) As Integer           'Lag
Dim VCT(6, 10) As Integer           'Critical Path Task Indicator 1/0
Dim VSE(6, 10) As Integer           'Square error
Dim VST(6, 10) As Integer           'Start Time for task j
Dim VFN(6, 10) As Integer           'Finish Time for task j
Dim VAMJ(6, 10) As Integer          'Assignment matrix Resource m to Task j
Dim VAHJ(6, 10) As Integer          'Assignment Matrix Skill level h to Tak j
Dim VAJT(6, 10, 800) As Integer     'Assignment resource m to task j in period t
'''End Declarations
```

```
Sub J10_Main()   'Main Programming Sequence for J10 (ten node projects) and K3 (three resource types)

Call Initialize                    '010

Call Robust_Scheduling_Method          '100

Call Project_Valuation_Method         '200

Call Project_Prioritization_Method    '300

Call Resource_Prioritization_Method    '400

For MR = KT To MRMAX

   Call Sequential_Scheduling_Method   '500

   Call Critical_Scheduling_Method   '600

   Call Early_Scheduling_Method      '700

   Call Level_Scheduling_Method       '800

   Call Level12_Assignment_Method    '900

   Call Level23_Assignment_Method    '1000

Next

Call Print_Chart


End Sub
```

```
Sub Initialize()

10: 'Initialize

JT = 10                 'Number of Tasks in each project
NP = 6                  'Number of Projects
MAX = 60                'Maximum number of resources
'MR = 24                'Number of Resources for particular run
MMR = 1                'Set to 1 if Multiple resource sets are run, prevents overflow
TP = 800               'Number of Time Periods
KT = 3                 'Number of Resource Types
LT = 3                 'Number of Resource Levels
Alpha = 0.1            'Confidence of regaining loss accross the portfolio
Beta = 0.1             'Likelyhood of further loss accross the portfolio
Delta = 0.2            'Near critical tasks are tasks with slack < delta*CP

Kappa1 = 0             'Workforce shaping, fraction of workforce as Type 1
Kappa2 = 0              'Workforce shaping, fraction of workforce as Type 2
Kappa3 = 0             'Workforce shaping, user defined fraction of workforce as
Type 3
                        'NOTE: Kappa1 + Kappa2+Kapp3 must equal 1
                        'If Kappa is 0, RGM automatically generates a workforce
                        'with distribution corresponding to work load demand
Lambda1 = 0             'Workforce shaping, fraction of workforce as Level 1
Lambda2 = 0            'Workforce shaping, fraction of workforce as Level 2
Lambda3 = 0             'Workforce shaping, fraction of workforce as Level 3
                        'NOTE: Lambda1 + Lambda2+Lambda3 must equal 1
                       'If Lamda is 0, RGM automatically generates a workforce
                       'with uniform skill level distribution

IR = 0.05 / 365        'Interest rate per day
Q1 = 0.3                'Level 1 rate, e.g. completes task 30% faster than level 2
Q3 = 0.3                'Level 3 rate, e.g. completes task 30% slower than level 2

PP = 0                 'Initialize project prioritization method,
                       '0)Sequential, 1)PRI index, 2) NPV, et

PR = 0                  'Resource Prioritization,
                        PR=0 Sequential, PR=1 Prioritized

RC1 = 1000              'Resource Level 1 Cost per day
RC2 = 800              'Rsource Level 2 Cost per day
RC3 = 500               'Resource Level 3 Cost per day


R = 1                  'Row Reference
```

```
C = 1                           'Column Reference
Row = 1                         'Row Assignment Output
Col = 1                          'Col Assignment Output
x = 1                            'row reference on Project worksheet
y = 1                           'column reference on Project worksheet
Excelrow = 1                     'row reference
Excelcol = 1                    'col reference
Resultsrow = 81                 'row reference in the results print procedure
testrow = 1                      'row reference


20: 'Projects Data
  x = 1
  y = 1
  For i = 1 To NP
    PI(i) = Worksheets("J10").Cells(x, y + 1)
    VP(i) = Worksheets("J10").Cells(x + 1, y + 1)
    TL(i) = Worksheets("J10").Cells(x + 2, y + 1)
    LP(i) = Worksheets("J10").Cells(x + 3, y + 1)
    For j = 1 To JT
      DM(i, j) = Worksheets("J10").Cells(x + 5, j + 1)
      DL(i, j) = Worksheets("J10").Cells(x + 6, j + 1)
      FJ(i, j) = Worksheets("J10").Cells(x + 7, j + 1)
      LG(i, j) = Worksheets("J10").Cells(x + 8, j + 1)
    Next
    x = x + 20
  Next

30: 'Read PRC from Projects into a two dimensional array
  x = 1
  For i = 1 To NP
    For jr = 1 To JT
      For jc = 1 To JT
        PRC(i, jr, jc) = Worksheets("J10").Cells(x + 9 + jr, jc + 1)
      Next
    Next
    x = x + 20
  Next

  'Print to make sure it has been initialized correctly
  x = 1
  y = 1
  For i = 1 To NP
    Worksheets("J10OUT").Cells(x, y + 1) = PI(i)
    Worksheets("J10OUT").Cells(x + 1, y + 1) = VP(i)
```

```
        Worksheets("J10OUT").Cells(x + 2, y + 1) = TL(i)
        Worksheets("J10OUT").Cells(x + 3, y + 1) = LP(i)
        For j = 1 To JT
            Worksheets("J10OUT").Cells(x + 5, j + 1) = DM(i, j)
            Worksheets("J10OUT").Cells(x + 6, j + 1) = DL(i, j)
            Worksheets("J10OUT").Cells(x + 7, j + 1) = FJ(i, j)
            Worksheets("J10OUT").Cells(x + 8, j + 1) = LG(i, j)
        Next
        x = x + 20
    Next

    'Print PRC to make sure it has been input correctly
    x = 1
    For i = 1 To NP
        For jr = 1 To JT
            For jc = 1 To JT
                Worksheets("J10OUT").Cells(x + 9 + jr, jc + 1) = PRC(i, jr, jc)
            Next
        Next
        x = x + 20
    Next


40: 'Initialize Critical Path/Critical Chain Variables
    For i = 1 To NP
        MAXEFPRC(i) = 0
        MINLSSUC(i) = 0
        MINESSUC(i) = 0
        NPVP(i) = 0
        RS(i) = 0
        For j = 1 To JT
            ES(i, j) = 0
            EF(i, j) = 0
            LS(i, j) = 0
            LF(i, j) = 0
            ST(i, j) = 0
            FN(i, j) = 0
            TS(i, j) = 0
            WS(i, j) = 0
            SC(i, j) = 0
            NC(i, j) = 0
        Next
    Next
```

```
' Resource Data File
'The following data input code is commented out, has been superseded by the Resource
'Generation Method and sub routine.  The code commented out here could be reinstated
'if a specific resource distribution is desired.


'For m = 1 To MAX            'index for resource number within group
   'For s = 1 To MAX         'index for size of group, M4, M8 etc.
      'z = (s * 2) + 1
      'GM(m, s) = Worksheets("K3").Cells(m + 1, s * 2)
      'HM(m, s) = Worksheets("K3").Cells(m + 1, z)
   'Next
'Next
'Print out GM and HM to verify input
'For m = 1 To MAX
   'For s = 1 To MAX
   'z = (s * 2) + 1
   'Worksheets("K3").Cells(50 + m, 2 * s).Value = GM(m, s)
   'Worksheets("K3").Cells(50 + m, z) = HM(m, s)
   'Next
'Next

99:   'End 010 - 99 Initialization

End Sub
'End Initialization


Sub Robust_Scheduling_Method()

100:   'Robust Scheduling Method (RSM)
      'Use Critical Path Method (CPM) and Critical Chain Buffer Management (CCBM)
      'to produce the unconstrained scheduling solution

      'Begin Critical Path Method

      'Initialize ES(1) and EF(1) for Forward Pass
105:   For i = 1 To NP
        ES(i, 1) = 0 + LG(i, 1)
        EF(i, 1) = DM(i, 1) + ES(i, 1)
      Next

      'Forward Pass
      'The earliest start (ES)for any task is equal to the maximum of the
      'earliest finish(EF)of the immediate predecessors
```

```
110:  For i = 1 To NP
        For j = 2 To JT 'start with task 2, search all rows in PRC for indicator
          ES(i, j) = 0
          For jr = 1 To JT
            If PRC(i, jr, j) = 1 Then 'it is a predecessor
              MAXEFPRC(i) = EF(i, jr) 'set it to the max constant
            End If
            If PRC(i, jr, j) = 1 And MAXEFPRC(i) >= ES(i, j) Then
              ES(i, j) = EF(i, jr)
              EF(i, j) = ES(i, j) + DM(i, j) + LG(i, j)
            End If
          Next
        Next
      Next

      'Initialize for Backward Pass
115:  For i = 1 To NP
        LF(i, JT) = EF(i, JT)
        LS(i, JT) = LF(i, JT) - DM(i, JT)
      Next
        'Backward Pass
      'The latest finish (LF) for any task is the smallest
      'of the latest start(LS) times of immediate successors
120:  For i = 1 To NP
        For j = JT - 1 To 1 Step -1
          LF(i, j) = LS(i, JT) '????
          For jr = JT To 1 Step -1
            If PRC(i, j, jr) = 1 Then
              MINLSSUC(i) = LS(i, jr)
            End If
            If PRC(i, j, jr) = 1 And MINLSSUC(i) <= LF(i, j) Then
              LF(i, j) = LS(i, jr)
              LS(i, j) = LF(i, j) - DM(i, j)
            End If
          Next
        Next
      Next

      'Compute Total Slack
      'Total Slack of activity j is the difference between 'the latest finish LF(J)
      'and earliest finish EF(J),or the difference between LS(J) and ES(J)
125:  For i = 1 To NP
        For j = 1 To JT
          TS(i, j) = LF(i, j) - EF(i, j)
        Next
      Next
      'Compute Free Slack(FS)
      'Free slack is the amount of time an activity may be 'delayed from the ES without
      'delaying the start of anyof it's successors.  Free slack is computed as the
      'difference between the min ES of the activity successors and the EF of the activity
130:   For i = 1 To NP
        For j = 1 To JT
        MINESSUC(i) = EF(i, JT)
          For jr = 1 To JT
            If PRC(i, j, jr) = 1 And ES(i, jr) <= MINESSUC(i) Then
              FS(i, j) = ES(i, jr) - EF(i, j)
              MINESSUC(i) = ES(i, j)
            End If
          Next
        Next
      Next


      'Compute Matrix of Critical Path Tasks
      'CT(J) = 1 if TS = 0, CT(J) = 0 if TS is not 0
135:   For i = 1 To NP
        For j = 1 To JT
          If TS(i, j) = 0 Then
            CT(i, j) = 1
          Else
            CT(i, j) = 0
          End If
        Next
      Next


      'Compute Critical Path Duration
      'CP is the sum of the durations d of CT =1
140:   For i = 1 To NP
        CP(i) = 0
        For j = 1 To JT
          If CT(i, j) = 1 Then
            CP(i) = CP(i) + DM(i, j) + LG(i, j)
          End If
        Next
      Next
```

```
     'Compute Matrix of Near Critical Path Tasks
     'Near critical task are tasks having slack < Delta * CP
145:   For i = 1 To NP
        For j = 1 To JT
          If CT(i, j) = 0 And TS(i, j) < Delta * CP(i) Then
            NC(i, j) = 1
          Else
            NC(i, j) = 0
          End If
        Next
     Next


     'Compute Square Root of Sum of Squares
150:   For i = 1 To NP
        For j = 1 To JT
          SE(i, j) = (DL(i, j) - DM(i, j)) ^ 2
        Next
     Next
     For i = 1 To NP
        SV(i) = 0
     Next
     For i = 1 To NP
        For j = 1 To JT
          SV(i) = SV(i) + (SE(i, j))
          SS(i) = Sqr(SV(i))
        Next
     Next

     'Compute Bias Buffer
155:   For i = 1 To NP
        BB(i) = 0.1 * CP(i)
     Next

     'Compute Critical Chain Duration,Equals expected duration TE
160:   For i = 1 To NP
        BF(i) = SS(i) + BB(i)
        CC(i) = CP(i) + BF(i)
        TE(i) = CC(i)
     Next


180:   ' Print Results from Unconstrained Scheduling Method
     Excelrow = 1
     Excelcol = 1
```

```
    Row = Excelrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "100 - ROBUST SCHEDULING
METHOD (RSM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "CRITICAL PATH/CRITICAL
CHAIN UNCONSTRAINED SCHEDULING"

    Row = Excelrow + 3
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "Project"
    Worksheets("Results").Cells(Row + 1, Col).Value = "Task"
    Worksheets("Results").Cells(Row + 2, Col).Value = "DM"
    Worksheets("Results").Cells(Row + 3, Col).Value = "DL"
    Worksheets("Results").Cells(Row + 4, Col).Value = "FJ"
    Worksheets("Results").Cells(Row + 5, Col).Value = "LG"
    Worksheets("Results").Cells(Row + 6, Col).Value = "ES"
    Worksheets("Results").Cells(Row + 7, Col).Value = "EF"
    Worksheets("Results").Cells(Row + 8, Col).Value = "LS"
    Worksheets("Results").Cells(Row + 9, Col).Value = "LF"
    Worksheets("Results").Cells(Row + 10, Col).Value = "TS"
    Worksheets("Results").Cells(Row + 11, Col).Value = "FS"
    Worksheets("Results").Cells(Row + 12, Col).Value = "CT"
    Worksheets("Results").Cells(Row + 13, Col).Value = "NC"
    Worksheets("Results").Cells(Row + 14, Col).Value = "CP"
    Worksheets("Results").Cells(Row + 15, Col).Value = "BF"
    Worksheets("Results").Cells(Row + 16, Col).Value = "CC"


    For i = 1 To NP
      For j = 1 To JT
        Worksheets("Results").Cells(Row, Col + 1).Value = i
        Worksheets("Results").Cells(Row + 1, Col + 1).Value = j
        Worksheets("Results").Cells(Row + 2, Col + 1).Value = DM(i, j)
        Worksheets("Results").Cells(Row + 3, Col + 1).Value = DL(i, j)
        Worksheets("Results").Cells(Row + 4, Col + 1).Value = FJ(i, j)
        Worksheets("Results").Cells(Row + 5, Col + 1).Value = LG(i, j)
        Worksheets("Results").Cells(Row + 6, Col + 1).Value = ES(i, j)
        Worksheets("Results").Cells(Row + 7, Col + 1).Value = EF(i, j)
        Worksheets("Results").Cells(Row + 8, Col + 1).Value = LS(i, j)
        Worksheets("Results").Cells(Row + 9, Col + 1).Value = LF(i, j)
        Worksheets("Results").Cells(Row + 10, Col + 1).Value = TS(i, j)
        Worksheets("Results").Cells(Row + 11, Col + 1).Value = FS(i, j)
        Worksheets("Results").Cells(Row + 12, Col + 1).Value = CT(i, j)
        Worksheets("Results").Cells(Row + 13, Col + 1).Value = NC(i, j)
        Col = Col + 1
      Next
      Next

      Col = Excelcol
      For i = 1 To NP
         Worksheets("Results").Cells(Row + 14, Col + 10).Value = CP(i)
         Worksheets("Results").Cells(Row + 15, Col + 10).Value = BF(i)
         Worksheets("Results").Cells(Row + 16, Col + 10).Value = CC(i)
         Col = Col + 10
      Next

199:   'End 100-199 Robust Scheduling Method (RSM)

End Sub


Sub Project_Valuation_Method()

200:   'Project Valuation Method (PVM)

205:   For i = 1 To NP
         DE(i) = TE(i) - TL(i)
         If DE(i) < 0 Then
            DE(i) = 0
         End If

         VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))
         VL(i) = VP(i) - VE(i)

         'Compute Potential Net Present Value NPVP
210:        GPL(i) = (4 * VP(i)) / (LP(i) ^ 2)
         GPR(i) = GPL(i)
         RPK(i) = (VP(i) / (LP(i) / 2))
         DFPGLP(i) = (((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2)))
         DFPFLP(i) = (1 + IR) ^ -TL(i)
         DFPARP(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
         DFPGRP(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
         DFPFRP(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
         NPVP(i) = GPL(i) * DFPGLP(i) * DFPFLP(i) + ((RPK(i) * DFPARP(i)) -
(GPR(i) * DFPGRP(i))) * DFPFRP(i)


         'Compute Expected NPV
215:        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
         REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
```

```
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1) /
((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
        DFPFLE(i) = (1 + IR) ^ -TE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
        NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

        'Compute NPL Lost
220:      NPVL(i) = NPVP(i) - NPVE(i)

     Next

     'Compute Total Net Present Value Expected TNPVE
225:   SUMNPVE = 0
     SUMNPVP = 1
     For ii = 1 To NP
        SUMNPVE = SUMNPVE + NPVE(ii)
        SUMNPVP = SUMNPVP + NPVP(ii)
     Next

     'Compute Program Schedule Efficiency
     PSE = SUMNPVE / SUMNPVP


240:   'Print Project Valuation Method Results
     Excelrow = 1
     Excelcol = 1
     Row = Excelrow + 22
     Col = Excelcol
     Worksheets("Results").Cells(Row, Col).Value = "200 - PROJECT VALUATION
METHOD (PVM)"
     Worksheets("Results").Cells(Row + 1, Col).Value = "BASED ON
UNCONSTRAINED SCHEDULES"
     Worksheets("Results").Cells(Row + 3, Col).Value = "Project"
     Worksheets("Results").Cells(Row + 4, Col).Value = "VP"
     Worksheets("Results").Cells(Row + 5, Col).Value = "TPL"
     Worksheets("Results").Cells(Row + 6, Col).Value = "LP"
     Worksheets("Results").Cells(Row + 7, Col).Value = "TE"
     Worksheets("Results").Cells(Row + 8, Col).Value = "DE"
     Worksheets("Results").Cells(Row + 9, Col).Value = "VE"
     Worksheets("Results").Cells(Row + 10, Col).Value = "VL"
     Worksheets("Results").Cells(Row + 11, Col).Value = "NPVP"
     Worksheets("Results").Cells(Row + 12, Col).Value = "NPVE"
     Worksheets("Results").Cells(Row + 13, Col).Value = "NPVL"
     Worksheets("Results").Cells(Row + 14, Col).Value = "Rate"
     Worksheets("Results").Cells(Row + 16, Col).Value = "TOTAL"
     Worksheets("Results").Cells(Row + 17, Col).Value = "NPVE"
     Worksheets("Results").Cells(Row + 18, Col).Value = "NPVP"
     Worksheets("Results").Cells(Row + 19, Col).Value = "PSE"
     Col = Excelcol
     For i = 1 To NP
        Worksheets("Results").Cells(Row + 3, Col + 10).Value = PI(i)
        Worksheets("Results").Cells(Row + 4, Col + 10).Value = VP(i)
        Worksheets("Results").Cells(Row + 5, Col + 10).Value = TL(i)
        Worksheets("Results").Cells(Row + 6, Col + 10).Value = LP(i)
        Worksheets("Results").Cells(Row + 7, Col + 10).Value = TE(i)
        Worksheets("Results").Cells(Row + 8, Col + 10).Value = DE(i)
        Worksheets("Results").Cells(Row + 9, Col + 10).Value = VE(i)
        Worksheets("Results").Cells(Row + 10, Col + 10).Value = VL(i)
        Worksheets("Results").Cells(Row + 11, Col + 10).Value = NPVP(i)
        Worksheets("Results").Cells(Row + 12, Col + 10).Value = NPVE(i)
        Worksheets("Results").Cells(Row + 13, Col + 10).Value = NPVL(i)
        Worksheets("Results").Cells(Row + 14, Col + 10).Value = IR
        Col = Col + 10
     Next
     Col = 1
     Worksheets("Results").Cells(Row + 17, Col + 1).Value = SUMNPVE
     Worksheets("Results").Cells(Row + 18, Col + 1).Value = SUMNPVP
     Worksheets("Results").Cells(Row + 19, Col + 1).Value = PSE


299:  'End 200-299 Project Valuation Method (PVM)

End Sub


Sub Project_Prioritization_Method()


300: 'Project Prioritization Method (PPM)

     'Rank order projects by NPVP
310:   For i = 1 To NP
        NPVPTEMP(i) = NPVP(i)
     Next

312:   For i = 1 To NP
        For j = 1 To NP - 1
```

```
          If NPVPTEMP(j) < NPVPTEMP(j + 1) Then
             NPTEMP = NPVPTEMP(j)
             NPVPTEMP(j) = NPVPTEMP(j + 1)
             NPVPTEMP(j + 1) = NPTEMP
          End If
       Next
    NPVPTEMPPRI(i) = i
    Next

314:   For i = 1 To NP  'Assign Priority
       For j = 1 To NP
          If NPVP(i) = NPVPTEMP(j) Then
             NPVPPRI(i) = NPVPTEMPPRI(j)
          End If
       Next
    Next   'Finished rank ordering NPVP


    'Rank order projects by VP Potential Value
320:   For i = 1 To NP
       VPTEMP(i) = VP(i)
    Next

322:   For i = 1 To NP
       For j = 1 To NP - 1
          If VPTEMP(j) < VPTEMP(j + 1) Then
             TEMP = VPTEMP(j)
             VPTEMP(j) = VPTEMP(j + 1)
             VPTEMP(j + 1) = TEMP
          End If
       Next
       VPTEMPPRI(i) = i
    Next

324:   For i = 1 To NP     'Assign Priority
       For j = 1 To NP
          If VP(i) = VPTEMP(j) Then
             VPRI(i) = VPTEMPPRI(j)
          End If
       Next
    Next   'Finished rank ordering by Potential Value


    'Rank order projects by NPVE
330:   For i = 1 To NP
       NPVETEMP(i) = NPVE(i)
    Next

332:  For i = 1 To NP
        For j = 1 To NP - 1
          If NPVETEMP(j) < NPVETEMP(j + 1) Then
            NETEMP = NPVETEMP(j)
            NPVETEMP(j) = NPVETEMP(j + 1)
            NPVETEMP(j + 1) = NETEMP
          End If
        Next
        NPVETEMPPRI(i) = i
    Next

334:  For i = 1 To NP     'Assign Priority
        For j = 1 To NP
          If NPVE(i) = NPVETEMP(j) Then
            NPVEPRI(i) = NPVETEMPPRI(j)
          End If
        Next
    Next
    'Finished rank ordering NPVE


    'Productivity Prioritization Method
    'Compute DVL/dt - derivative of the value lost - loss rate factor
340:  For i = 1 To NP
        DV(i) = (VP(i) / LP(i)) * (3 - ((4 * DE(i)) / LP(i)))
    Next

    'Find the maximum loss rate
342:    MAXDV = Application.WorksheetFunction.MAX(DV(1), DV(2), DV(3), DV(4),
DV(5), DV(6))
    'Compute the Project Productivity Metric (PPM)

344:  For i = 1 To NP
        GAMMA(i) = DV(i) / MAXDV
        PPM(i) = ((NPVE(i) + (Alpha * NPVL(i))) / TE(i)) * (1 + Beta * GAMMA(i))
    Next

346:  For i = 1 To NP     'Rank order projects by PPM
        PTEMP(i) = PPM(i)
    Next

348:  For i = 1 To NP
        For j = 1 To NP - 1
          If PTEMP(j) < PTEMP(j + 1) Then
```

```
                TEMP = PTEMP(j)
                PTEMP(j) = PTEMP(j + 1)
                PTEMP(j + 1) = TEMP
            End If
        Next
        PRITEMP(i) = i
    Next

350:  For i = 1 To NP 'Assign PRI Priority
        For j = 1 To NP
            If PPM(i) = PTEMP(j) Then
                PRI(i) = PRITEMP(j)
            End If
        Next
    Next


    'Random Rank Ordering for Projects
360:   For i = 1 To NP
        Randomize
        RA(i) = Rnd
        RB(i) = RA(i)
    Next
362:  For i = 1 To NP
        For j = 1 To NP - 1
            If RB(j) < RB(j + 1) Then
                RTEMP = RB(j)
                RB(j) = RB(j + 1)
                RB(j + 1) = RTEMP
            End If
        Next
        PRB(i) = i
    Next
364:  For i = 1 To NP
        For j = 1 To NP
            If RA(i) = RB(j) Then
                RPRI(i) = PRB(j)
            End If
        Next
    Next


370:  'Early Start Rank Ordering of Tasks, total tasks = NP*JT
    NJT = NP * JT
```

```
      k = 0
      STEP = 0.001
      'Initialize the vector of Early Starts from 1 to NJT
      'Add STEP to every value so that no to values are the same
372:  For i = 1 To NP
        For j = 1 To JT
          k = k + 1
          ESA(k) = ES(i, j) + STEP
          STEP = STEP + 0.001
        Next
      Next


      'Add Step where ever ES=0
      'k = 0
      'kk = 0
374:  'For i = 1 To NP
        'For j = 1 To JT
          'k = k + 1
          'If ES(i, j) = 0 Then
            'ESA(k) = ES(i, j) + STEP
            'STEP = STEP + 0.001
          'End If
        'Next
      'Next
      'Add Step where ever two values are equal
375:    'For i = 1 To NP
        'For j = 1 To JT
          'kk = 0
          'For ii = 1 To NP
            'For jj = 1 To JT
              'kk = kk + 1
              'If ES(i, j) = ES(ii, jj) Then
                'ESA(kk) = ES(ii, jj) + STEP
                'STEP = STEP + 0.001
              'End If
            'Next
          'Next
        'Next
      'Next


      'Initialize sorting array ESB
      k = 0
376:  For i = 1 To NP
        For j = 1 To JT
```

```
            k = k + 1
            ESB(k) = ESA(k)
        Next
    Next


377:  For i = NJT To 1 Step -1
        For j = 1 To i - 1
            If ESB(j) > ESB(j + 1) Then
                RTEMP = ESB(j)
                ESB(j) = ESB(j + 1)
                ESB(j + 1) = RTEMP
            End If
        Next
        PESB(i) = i
    Next

378:  For i = 1 To NJT
        For j = 1 To NJT
            If ESA(i) = ESB(j) Then
                PESA(i) = PESB(j)
            End If
        Next
    Next


    For k = 1 To NJT
        For p = 1 To NJT
            If k = PESA(p) Then
                ESARO(k) = ESA(p)
            End If
        Next
    Next


390:  'Print Project Ranking Method Results
    Excelrow = 1
    Excelcol = 1
    Row = Excelrow + 44
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "300 - PROJECT
PRIORITIZATION METHOD (PPM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "FOR RESOURCE
ASSIGNMENT"
    Worksheets("Results").Cells(Row + 3, Col).Value = "SPI - Sequential Project Index
(User Defined)"
```

```
    Worksheets("Results").Cells(Row + 5, Col).Value = "NPI - NPV Project Index"
    'Worksheets("Results").Cells(Row + 7, Col).Value = "VPI - Value Project Index"
    Worksheets("Results").Cells(Row + 8, Col).Value = "Project Productivity Ranking
Method"
    Worksheets("Results").Cells(Row + 9, Col).Value = "Alpha"
    Worksheets("Results").Cells(Row + 10, Col).Value = "Beta"
    Worksheets("Results").Cells(Row + 11, Col).Value = "DVLDT"
    Worksheets("Results").Cells(Row + 12, Col).Value = "MAXDV"
    Worksheets("Results").Cells(Row + 13, Col).Value = "GAMMA"
    Worksheets("Results").Cells(Row + 14, Col).Value = "PPM"
    Worksheets("Results").Cells(Row + 15, Col).Value = "PPI - Productivity Project
Index "
    Worksheets("Results").Cells(Row + 17, Col).Value = "Random Number"
    Worksheets("Results").Cells(Row + 18, Col).Value = "RPI - Random Project Index
"
    Worksheets("Results").Cells(Row + 21, Col).Value = "ES"
    Worksheets("Results").Cells(Row + 22, Col).Value = "Index k"
    Worksheets("Results").Cells(Row + 23, Col).Value = "ESA "
    Worksheets("Results").Cells(Row + 24, Col).Value = "PESA"
    Worksheets("Results").Cells(Row + 25, Col).Value = "ESARO"

    Col = Excelcol
391:  For i = 1 To NP

      Worksheets("Results").Cells(Row + 3, Col + 10).Value = PI(i)
      Worksheets("Results").Cells(Row + 5, Col + 10).Value = NPVPPRI(i)
      'Worksheets("Results").Cells(Row + 7, Col + 10).Value = VPRI(i)
      Worksheets("Results").Cells(Row + 9, Col + 10).Value = Alpha
      Worksheets("Results").Cells(Row + 10, Col + 10).Value = Beta
      Worksheets("Results").Cells(Row + 11, Col + 10).Value = DV(i)
      Worksheets("Results").Cells(Row + 12, Col + 10).Value = MAXDV
      Worksheets("Results").Cells(Row + 13, Col + 10).Value = GAMMA(i)
      Worksheets("Results").Cells(Row + 14, Col + 10).Value = PPM(i)
      Worksheets("Results").Cells(Row + 15, Col + 10).Value = PRI(i)

      Worksheets("Results").Cells(Row + 17, Col + 10).Value = RA(i)
      Worksheets("Results").Cells(Row + 18, Col + 10).Value = RPRI(i)

      Col = Col + 10
    Next

    Col = 1
    For i = 1 To NP
      For j = 1 To JT
        Worksheets("Results").Cells(Row + 21, Col + 1).Value = ES(i, j)
        Col = Col + 1
      Next
    Next

    Col = 1

    For k = 1 To NJT
      Worksheets("Results").Cells(Row + 22, Col + k).Value = k
      Worksheets("Results").Cells(Row + 23, Col + k).Value = ESA(k)
      Worksheets("Results").Cells(Row + 24, Col + k).Value = PESA(k)
      Worksheets("Results").Cells(Row + 25, Col + k).Value = ESARO(k)
    Next


392:  ' Print Resource Prioritization Method
    'Col = 1
    'Worksheets("Results").Cells(Row + 27, Col).Value = "400 - RESOURCE
PRIORITIZATION METHOD (RPM)"
    'Worksheets("Results").Cells(Row + 28, Col).Value = "SRI - Sequential Resource
Index (User Defined)"
    'Worksheets("Results").Cells(Row + 29, Col).Value = "GM - Resource Type"
    'Worksheets("Results").Cells(Row + 30, Col).Value = "HM - Skill Level"
    'Worksheets("Results").Cells(Row + 31, Col).Value = "LRI - Level Resource Index"

    'Worksheets("M16OUT").Cells(Row + 33, Col).Value = "RRI - Random Resource
Index"
    'Col = 10
    'For m = 1 To MR
      'Worksheets("Results").Cells(Row + 28, Col + m).Value = m
      'Worksheets("Results").Cells(Row + 29, Col + m).Value = GM(m, MR)
      'Worksheets("Results").Cells(Row + 30, Col + m).Value = HM(m, MR)
      'Worksheets("Results").Cells(Row + 31, Col + m).Value = LRI(m, MR)
      'Worksheets("Results").Cells(Row + 33, Col + m).Value = RRI(m)

    'Next

399:  'End 300-399 Project Prioritization Method (PPM)

End Sub
```

```
Sub Resource_Prioritization_Method()

400:  'Resource Prioritization Method (RPM)
    'This procedure generates a set of resource sets sized from the
    'minimum (KT=No. Types) to a maximum number incremented by one.
    'The resource sets are used to test performance of the various
    'scheduling and assignment algorithms using a range of work
    'force sizes, types and skill levels.'Resource sets are developed
    'using project demand requirements and parameters that allow the
    'user to attain a distribution of esource types and skill levels
    'representative of an actual work force.

401:  'Local variable declaration
    Dim RK1 As Integer        'Requirements Resource Type 1
    Dim RK2 As Integer        'Requirements Resource Type 2
    Dim RK3 As Integer        'Requirements Resource Type 3
    Dim RKT As Integer        'Total resource requirements

    Dim DK1 As Variant        'Demand for Resource Type 1
    Dim DK2 As Variant        'Demand for Resource Type 2
    Dim DK3 As Variant        'Demand for Resource Type 3

    Dim QK1 As Integer        'Available Quant of Resource Type 1
    Dim QK2 As Integer        'Available Quant of Resource Type 2
    Dim QK3 As Integer        'Available Quant of Resource Type 3
    Dim QKT As Integer        'Available Quant of Total Resources

    Dim AK1 As Variant        'Available Ratio Resource Type 1
    Dim AK2 As Variant        'Available Ratio Resource Type 2
    Dim AK3 As Variant        'Available Ratio Resource Type 3

    Dim CK1 As Integer        'Coinciding Resources Type 1
    Dim CK2 As Integer        'Coinciding Resources Type 2
    Dim CK3 As Integer        'Coinciding Resources Type 3
    Dim CKT As Integer        'Total Coinciding Resources

    Dim LL1 As Integer        'Available Skill Level 1
    Dim LL2 As Integer        'Available Skill Level 2
    Dim LL3 As Integer        'Available Skill Level 3
    Dim LLT As Integer        'Available Skill Level Total

    Dim DF(3) As Variant      'Resource Type Deficiency
    Dim LD(3) As Variant      'Resource Skill Level Deficiency

    Dim TYPEADD As Integer    'Resource Type to be Added
    Dim LEVELADD As Integer   'Resource Skill Level to be Added
    Dim TGM(60, 60) As Variant  'Temp Resource Type for sorting
    Dim THM(60, 60) As Variant  'Temp Skill Level for sorting

    Dim TLRI(60, 60) As Integer 'Resource Priority Index for sorting
    Dim AM(60, 60) As Variant   'Resource Type Matrix for sorting
    Dim BM(60, 60) As Variant   'Resource Type Matrix for sorting

    Dim TEMPINT1 As Integer      'Temp Value for sorting
    Dim TEMPINT2 As Integer      'Temp Value for sorting

    Dim TEMP1 As Variant        'Temp Value for sorting
    Dim TEMP2 As Variant        'Temp Value for sorting


402:  'Initialization
    RK1 = 0
    RK2 = 0
    RK3 = 0
    RKT = 1

    LEVELADD = 0

    CK1 = 0
    CK2 = 0
    CK3 = 0
    CKT = 0

    CK1MAX = 0
    CK2MAX = 0
    CK3MAX = 0
    MRMAX = 0

    For n = 1 To 60
      RC(n) = 0
      CF(n) = 0
    Next

    'Determine Maximum number of Resources Needed to be Generated
    'by determining the number of coincidental resources for each type
    'and then summing each type.
405:  For i = 1 To NP
406:      For j = 1 To JT
            CK1 = 0
            CK2 = 0
            CK3 = 0
```

```
407:    If FJ(i, j) = 1 Then

408:      For ii = 1 To NP

409:        For jj = 1 To JT

410:          If i = ii And j = jj Then
               CK1 = CK1
               Else

412:             If FJ(ii, jj) = 1 Then
                  If ES(i, j) <= EF(ii, jj) And EF(i, j) >= ES(ii, jj) Then
                    CK1 = CK1 + 1
                    If CK1 > CK1MAX Then
                      CK1MAX = CK1
                    End If
                  End If
               End If

            End If
          Next
        Next

414:    ElseIf FJ(i, j) = 2 Then

416:      For ii = 1 To NP

418:        For jj = 1 To JT

420:          If i = ii And j = jj Then
              CK2 = CK2

            Else

422:             If FJ(ii, jj) = 2 Then
                  If ES(i, j) <= EF(ii, jj) And EF(i, j) >= ES(ii, jj) Then
                    CK2 = CK2 + 1

                    If CK2 > CK2MAX Then
                      CK2MAX = CK2

                    End If
                  End If
                End If
             End If
            Next
          Next

424:     ElseIf FJ(i, j) = 3 Then
          CK3 = CK3 + 1

426:        For ii = 1 To NP

428:          For jj = 1 To JT

430:            If i = ii And j = jj Then
                 CK3 = CK3

              Else

432:              If FJ(ii, jj) = 3 Then

                   If ES(i, j) <= EF(ii, jj) And EF(i, j) >= ES(ii, jj) Then
                     CK3 = CK3 + 1
                     If CK3 > CK3MAX Then
                       CK3MAX = CK3
                     End If
                   End If
                End If  '432

              End If     '430
            Next         '428
          Next           '426
        End If           '407
      Next               '406
    Next                 '405


435:  'Determine Maximum Resource Required
    MRMAX = CK1MAX + CK2MAX + CK3MAX
    If MRMAX < NP * JT Then
      MRMAX = MRMAX + ((NP * JT) - MRMAX) / 2
    End If

    MR = MRMAX


    ' Determine the total amount of days required by each type of resource.
440:  For i = 1 To NP

441:    For j = 2 To JT - 1
```

```
               If FJ(i, j) = 1 Then
                  RK1 = RK1 + DM(i, j)
               ElseIf FJ(i, j) = 2 Then
                  RK2 = RK2 + DM(i, j)
               ElseIf FJ(i, j) = 3 Then
                  RK3 = RK3 + DM(i, j)
               End If

            Next   '441 End
         Next      '440 End

         'Compute Total Resource Days Required for the Program
   442:  RKT = RK1 + RK2 + RK3


   444:  'Compute demand percent for each type
         DK1 = RK1 / RKT
         DK2 = RK2 / RKT
         DK3 = RK3 / RKT


         'Initialize Resource Type matrix and Skill Level Matrix to zero
   448:  For n = KT To MRMAX
            For m = 1 To n      'index for size of group, M4, M8 etc.
               GM(m, n) = 0
               HM(m, n) = 0
            Next
         Next

         'Initialize Resource Group, one of each TYPE 1,2 3, all LEVEL 2
   450:  n = KT
         For m = 1 To n
            GM(m, n) = m
            HM(m, n) = 2
         Next

         'Generate Addition Resource Sets
         z = 4
   452:  For n = 3 To MRMAX - 1

            'Populate next set with same type and level to m
   454:        For m = 1 To n
               GM(m, n + 1) = GM(m, n)
               HM(m, n + 1) = HM(m, n)
            Next
```

```
        'Compute resource TYPE quantity currently available for each set
        ' QK1 type 1 avail, QK2 type 2 avail, QK3 type 3 avail
        QK1 = 0
        QK2 = 0
        QK3 = 0

        'Quantity of each Resource Type Available
456:      For m = 1 To n
           If GM(m, n) = 1 Then
              QK1 = QK1 + 1
           ElseIf GM(m, n) = 2 Then
              QK2 = QK2 + 1
           ElseIf GM(m, n) = 3 Then
           QK3 = QK3 + 1
           End If
        Next


        'Total resources available
458:      QKT = QK1 + QK2 + QK3

460:      'Compute available Type ratio
        AK1 = QK1 / QKT
        AK2 = QK2 / QKT
        AK3 = QK3 / QKT

        'Compute resource type difficiency for each type, (Resource Type Demand RKx -
Available Resource ARx)
462:      For i = 1 To KT

464:         If i = 1 Then
              If Kappa1 > 0 Then
                 DF(i) = Kappa1 - AK1
              Else
                 DF(i) = DK1 - AK1
              End If
           End If

466:         If i = 2 Then
              If Kappa2 > 0 Then
                 DF(i) = Kappa2 - AK2
              Else
                 DF(i) = DK2 - AK2
              End If
           End If
```

```
468:          If i = 3 Then
                If Kappa3 > 0 Then
                  DF(i) = Kappa3 - AK3
                Else
                  DF(i) = DK3 - AK3
                End If
              End If
            Next   '462 End


            'Test and assign resource type with the maximum difference to be added
470:          If DF(1) > DF(2) And DF(1) > DF(3) Then     'Case with DF(1) max
              TYPEADD = 1

              ElseIf DF(2) > DF(1) And DF(2) > DF(3) Then     'Case with DF(2) max
                TYPEADD = 2

              ElseIf DF(3) > DF(1) And DF(3) > DF(2) Then     'Case with DF(3) max
              TYPEADD = 3

              ElseIf DF(1) = DF(2) Or DF(1) = DF(3) Or DF(2) = DF(3) Then  'Tie breaker
                TYPEADD = 2
            End If

            'Shape SKILL LEVEL distribution
            'If no Lamdas are assigned, the generator will develop a uniform distribution
472:          If Lambda1 > 0 Or Lambda2 > 0 Or Lambda3 > 0 Then

474:            'Compute SKILL LEVEL quantity available for each set
              LL1 = 0
              LL2 = 0
              LL3 = 0

476:            For m = 1 To n
                If HM(m, n) = 1 Then
                  LL1 = LL1 + 1
                ElseIf HM(m, n) = 2 Then
                  LL2 = LL2 + 1
                ElseIf HM(m, n) = 3 Then
                  LL3 = LL3 + 1
                End If
              Next

478:            'Total Skill Levels
              LLT = LL1 + LL2 + LL3
```

```
480:       'Compute ratios of Available SKILL LEVELS
          AL1 = LL1 / LLT
          AL2 = LL2 / LLT
          AL3 = LL3 / LLT

481:       'Compute Level deficiency
          LD(1) = Lambda1 - AL1
          LD(2) = Lambda2 - AL2
          LD(3) = Lambda3 - AL3

          'If Lambdas positive, test and assign skill level with the maximum difference to
be added
482:       If LD(1) > LD(2) And LD(1) > LD(3) Then    'Case with LD(1) max
           LEVELADD = 1

           ElseIf LD(2) > LD(1) And LD(2) > LD(3) Then    'Case with LD(2) max
             LEVELADD = 2

           ElseIf LD(3) > LD(1) And LD(3) > LD(2) Then    'Case with LD(3) max
             LEVELADD = 3

           ElseIf LD(1) = LD(2) Or LD(1) = LD(3) Or LD(2) = LD(3) Then  'Tie
breaker
             LEVELADD = 2
          End If
        Else    '472 Else


484:      'Level Increment
        LEVELADD = LEVELADD + 1

        End If  '472 End


485:      'Add a resource
        GM(n + 1, n + 1) = TYPEADD
        HM(n + 1, n + 1) = LEVELADD
        If LEVELADD = 3 Then
          LEVELADD = 0
        End If

      Next     '452 End Iterate from n = 3 to MR-1 - All Resource Sets


      'SORT all Resources in Ascending order of Skill Level, Highest priority to lowest
```

```
     'Introduce non-integer variable AM corresponding to GM and HM
     'Each value of AM will be unique by adding a STEP = 0.001
     'BM will be initiated for sorting.
     STEP = 0.001
486:   For n = 3 To MRMAX
         For m = 1 To n
           AM(m, n) = HM(m, n) + STEP
           BM(m, n) = AM(m, n)
           STEP = STEP + 0.001
           TGM(m, n) = GM(m, n)  'Initialize temporary sorting matrices for prioritizing
           THM(m, n) = HM(m, n)

         Next
     Next

     'This sort for ascending order
487:   For n = 3 To MRMAX
488:     For i = n To 1 Step -1
489:       For j = 1 To i - 1
           If BM(j, n) > BM(j + 1, n) Then
               TEMP1 = BM(j, n)
               TEMPINT1 = TGM(j, n)
               TEMPINT2 = THM(j, n)
               BM(j, n) = BM(j + 1, n)
               THM(j, n) = THM(j + 1, n)
               TGM(j, n) = TGM(j + 1, n)
               BM(j + 1, n) = TEMP1
               THM(j + 1, n) = TEMPINT1
               TGM(j + 1, n) = TEMPINT2
             End If
           Next
           TLRI(i, n) = i
         Next
     Next

     'Assign Priorities
490:   For n = 3 To MRMAX
         For i = 1 To n
           For j = 1 To n
             If AM(i, n) = BM(j, n) Then
               LRI(i, n) = TLRI(j, n)
             End If
           Next
         Next
     Next
```

```
491:  'Print Prioritizes GM and HM
     Excelrow = 1
     Excelcol = 1

     'Print row heading, 1 to MR max
     Row = Excelrow
     Col = Excelcol
     For m = 1 To MRMAX
        Worksheets("K3O").Cells(Row + 2, Col).Value = m
        Row = Row + 1
     Next

     Row = Excelrow
     Col = Excelcol
     'Print Col Headings
     For S = 1 To MRMAX
        z = (S * 2) + 1
        Worksheets("K3O").Cells(Row, 2 * S).Value = S
        Worksheets("K3O").Cells(Row + 1, 2 * S).Value = "GM"
        Worksheets("K3O").Cells(Row + 1, z) = "HM"
     Next

492:  'Print GM and HM if PP=0
     Row = Excelrow + 1
     For n = 3 To MRMAX
        For m = 1 To n
           z = (n * 2) + 1
           Worksheets("K3O").Cells(Row + m, 2 * n).Value = GM(m, n)
           Worksheets("K3O").Cells(Row + m, z) = HM(m, n)
        Next
     Next

     Row = 60
     'Print out priority Index
     For n = 3 To MRMAX
        For m = 1 To n
           Worksheets("K3O").Cells(Row + m, 2 * n).Value = LRI(m, n)
        Next
    Next

     'If PP = 1, Prioritize GM and HM
493:    If PR = 1 Then     'If PP=1, arrange GM and HM according to priority order


        'Populate temporary sorting matrices
    For n = 3 To MRMAX
       For m = 1 To n
          THM(m, n) = HM(m, n)
          TGM(m, n) = GM(m, n)
       Next
    Next

    'Sort by LRI

    For n = 3 To MRMAX
       For i = 1 To n
          For m = 1 To n
             j = LRI(m, n)
             If i = m Then
                HM(j, n) = THM(i, n)
                GM(j, n) = TGM(i, n)
             End If
          Next
       Next
    Next
 End If  '493 End


494:  'Print Prioritized GM and HM

 Excelrow = 120
 Excelcol = 1

 'Print row heading, 1 to MR max
 Row = Excelrow
 Col = Excelcol
 For m = 1 To MRMAX
    Worksheets("K3O").Cells(Row + 2, Col).Value = m
    Row = Row + 1
 Next

 Row = Excelrow
 Col = Excelcol
 'Print Col Headings
 For S = 1 To MRMAX
    z = (S * 2) + 1
    Worksheets("K3O").Cells(Row, 2 * S).Value = S
    Worksheets("K3O").Cells(Row + 1, 2 * S).Value = "GM"
    Worksheets("K3O").Cells(Row + 1, z) = "HM"
 Next
```

```
      'Print GM and HM if PP=0
      Row = Excelrow + 1
      For n = 3 To MRMAX
         For m = 1 To n
            z = (n * 2) + 1
            Worksheets("K3O").Cells(Row + m, 2 * n).Value = GM(m, n)
            Worksheets("K3O").Cells(Row + m, z) = HM(m, n)
         Next
      Next


      'Determine Resource Cost of each resource set
495:    For n = KT To MRMAX
         For m = 1 To n
            If HM(m, n) = 1 Then
               RC(n) = RC(n) + RC1
            ElseIf HM(m, n) = 2 Then
               RC(n) = RC(n) + RC2
            ElseIf HM(m, n) = 3 Then
               RC(n) = RC(n) + RC3
            End If
         Next
         CF(n) = 2 * (200 * RC(n)) / SUMNPVE 'Determine Fractional Cost, RC/NPVP
      Next


498:    ' Print Resource Prioritization Method
      Row = 72
      Col = 1
      Worksheets("Results").Cells(Row, Col).Value = "400 - RESOURCE
PRIORITIZATION METHOD (RPM)"
      Worksheets("Results").Cells(Row + 1, Col).Value = "SRI - Sequential Resource
Index"
      Worksheets("Results").Cells(Row + 2, Col).Value = "GM - Resource Type"
      Worksheets("Results").Cells(Row + 3, Col).Value = "HM - Skill Level"
      Worksheets("Results").Cells(Row + 4, Col).Value = "LRI - Level Resource Index"

      'Worksheets("M16OUT").Cells(Row + 5, Col).Value = "RRI - Random Resource
Index"

      Col = 10
      For m = 1 To MR
         Worksheets("Results").Cells(Row + 1, Col + m).Value = m
         Worksheets("Results").Cells(Row + 2, Col + m).Value = GM(m, MR)
         Worksheets("Results").Cells(Row + 3, Col + m).Value = HM(m, MR)
         Worksheets("Results").Cells(Row + 4, Col + m).Value = LRI(m, MR)
         'Worksheets("Results").Cells(Row + 5, Col + m).Value = RRI(m)
      Next

499:   'End 400-499 Resource Prioritization Method (RPM)

End Sub


Sub Sequential_Scheduling_Method()

500:   'Sequential Scheduling Method(SSM) is a multi-pass assignment algorithm.
      'The algorithms assigns tasks in order of project priority and sequential
      'task number using two forward passes.

      '1) The first pass develops a TIME CONSTRAINED SOLUTION (TCS).
      'A TCS is developed within time constraints for tasks and projects.
      'If a TCS project contains some tasks unassigned then that project
      'is assumed to have insufficient resources to meet time
      'constraints and therefore not valued in the TCS.

      '2) The second pass develops a RESOURCE CONSTRAINED SOLUTION (RCS).
      'Starting with the TCS, projects not fully resourced are identified
      'and assignments on those projects are reset to zero. These projects
      'not fully resourced are then scheduled by incrementally relaxing
      'date constraints and by forcing an assignment to each task by adding
      'delay to the tasks and to the projects.

501:   'Local Variable Declarations
      Dim TES(6, 10) As Integer  'Temporary Early Start
      Dim TEF(6, 10) As Integer  'Temporary Early Finish
      Dim TLS(6, 10) As Integer  'Temporary Early Start
      Dim TLF(6, 10) As Integer  'Temporary Early Finish
      Dim TTS(6, 10) As Integer  'Temporary Slack


502:   'Local Variable Initialization
      For i = 1 To NP
         For j = 1 To JT
            TES(i, j) = ES(i, j)
            TEF(i, j) = EF(i, j)
            TLS(i, j) = LS(i, j)
            TLF(i, j) = LF(i, j)
            TTS(i, j) = TS(i, j)
         Next
```

```
       Next

505:   'Initialize Variable and Assignments
       For i = 1 To NP
          RS(i) = 0
          For j = 1 To JT
             AMJ(i, j) = 0
             ST(i, j) = 0
             FN(i, j) = 0
             WS(i, j) = TS(i, j)
             SC(i, j) = 0
             LG(i, j) = 0

             For t = 1 To TP
                AJT(i, j, t) = 0
             Next
          Next
       Next


510:   'Time Constrained Solution (TCS)

       'Select Priority Order for Projects
511:   For n = 1 To NP
          If PP = 0 Then         'PP=0 for Sequentially ordered projects
             i = n
          End If
          If PP = 1 Then         'PP=1 for PRI Productivity prioritized projects
             i = PRI(n)
          End If
          If PP = 2 Then         'PP=2 for NPVP Potential NPV prioritized projects
             i = NPVPPRI(n)
          End If
          If PP = 3 Then          'PP=3 for Random Project Order
             i = RPRI(n)
          End If

             'Assign Tasks one at a time
512:          For j = 2 To JT - 1
                ST(i, j) = TES(i, j) + 1 'Start at period ES +1


                'Assign Resources on at a time
513:               For m = 1 To MR  'Try to assign resources sequentially

                   'Check for Resource Type match
```

```
514:              If FJ(i, j) = GM(m, MR) Then

                  'Check if resource is available during period
                  MA(i, j) = 0
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                     For ii = 1 To NP
                        For jj = 1 To JT
                           If AJT(ii, jj, t) = m Then
                              MA(i, j) = 1 'set to 1 if not avail
                           End If
                        Next
                     Next
                  Next

                  'Assign if available
                  If MA(i, j) = 0 Then
                     AMJ(i, j) = m
                     FN(i, j) = ST(i, j) + DM(i, j) - 1

                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        AJT(i, j, t) = m 'assign m to task k in period t
                     Next
                     GoTo NEXTTASK 'Redirect to next task
                  End If
               End If     '514 FJ=GM
            Next          '513 Try all resources directly


            'If no direct assignment, delay using slack if available
515:           Do While SC(i, j) < TS(i, j)

               SC(i, j) = SC(i, j) + 1
               WS(i, j) = TS(i, j) - SC(i, j)
               ST(i, j) = TES(i, j) + SC(i, j)

               'Assign Resources one at a time
516:             For m = 1 To MR

                  'Check for Resource Type Match
517:                If FJ(i, j) = GM(m, MR) Then

                     'Check if resource is avail during period
                     MA(i, j) = 0
                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        For ii = 1 To NP
                           For jj = 1 To JT
```

```
                      If AJT(ii, jj, t) = m Then
                        MA(i, j) = 1 'set to 1 if not avail
                      End If

                    Next
                  Next
                Next

                'If m is available in required interval, assign to j
518:              If MA(i, j) = 0 Then

                  AMJ(i, j) = m 'Assign resource m to proj i task j
                  FN(i, j) = ST(i, j) + DM(i, j) - 1

                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    AJT(i, j, t) = m 'assign m to task k in period t
                  Next


                  'Forward Pass to Update TES, TEF
                  MAXEFPRC(i) = 0
519:               For jj = j To JT 'start with task 2
                    TES(i, jj) = ST(i, jj)
                    For jr = 1 To JT
                      If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                        MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                      End If
                      If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                        TES(i, jj) = TEF(i, jr)
                        TEF(i, jj) = TES(i, jj) + DM(i, jj)
                      End If
                    Next
                  Next

                  GoTo NEXTTASK 'Redirect to next task

                End If '518
              End If    '517 FJ=GM
            Next        '516 Try next resource m
          Loop          '515 Loop while slack is available
NEXTTASK:
        Next      '512 Iterate for JT tasks

      Next        '511 Iterate for NP projects
```

```
    'Last node has TES/TEF equal to largest of TEF of predecessors
520: For i = 1 To NP
        MAXTEFPRC(i) = 0
         For jr = 1 To JT
            If PRC(i, jr, JT) = 1 And TEF(i, jr) >= MAXTEFPRC(i) Then
               MAXTEFPRC(i) = TEF(i, jr) 'set it to the max constant
            End If
         Next
     Next


     'Complete Sequential Scheduling Method - Time Constrained Solution
522:  For i = 1 To NP
        ST(i, JT) = MAXTEFPRC(i)     'Last node start equals SES (updated)
        FN(i, JT) = MAXTEFPRC(i)     'Last node finish equals SEF (updated)

        'If not resourced, set schedule/value to null
        For j = 2 To JT - 1
           If AMJ(i, j) = 0 Then
              ST(i, JT) = 0
              FN(i, JT) = 0
              BF(i) = 0
              DE(i) = 0
           End If

        'Otherwise complete the schedule
        Next
        CP(i) = FN(i, JT)
        TE(i) = CP(i) + BF(i)
        DE(i) = TE(i) - TL(i)
        If DE(i) < 0 Then
           DE(i) = 0
        End If
     Next


     'Compute VE and NPVE for the projects that are scheduled
525:  For i = 1 To NP
        VE(i) = 0
        NPVE(i) = 0
        If TE(i) > 0 Then
           VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))

           'Compute Expected NPV
           GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
           REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
```

```
            GER(i) = REK(i) / (LP(i) / 2)
            DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1)
/ ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
            DFPFLE(i) = (1 + IR) ^ -TE(i)
            DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
            DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
            DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
            NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

         End If
      Next

      'Compute Total Net Present Value Expected TNPVE
526:   SUMNPVE = 1
      For ii = 1 To NP
         SUMNPVE = SUMNPVE + NPVE(ii)
      Next

      'Compute Program Schedule Efficiency
527:   PSE = SUMNPVE / SUMNPVP
      SSMTCS(MR) = PSE

   'Format TCS Variables for printing, this preserves the original solution
530:  If MMR = 0 Then          'Print only if a single resource set is run

     PSEO = PSE
     SUMNPVEO = SUMNPVE
     SUMNPVPO = SUMNPVP

     For i = 1 To NP
        CPO(i) = CP(i)
        BFO(i) = BF(i)
        TEO(i) = TE(i)
        DEO(i) = DE(i)
        NPVEO(i) = NPVE(i)

        For j = 1 To JT
           DMO(i, j) = DM(i, j)
           DLO(i, j) = DL(i, j)
           FJO(i, j) = FJ(i, j)
           ESO(i, j) = ES(i, j)
           EFO(i, j) = EF(i, j)
           LSO(i, j) = LS(i, j)
           LFO(i, j) = LF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


535:  'Print Results; Time; CONSTRAINED; SOLUTION(TCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "500 - SEQUENTIAL
SCHEDULING METHOD(SSM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "TIME CONSTRAINED
SOLUTION (TCS)"
    Row = Row + 2

    Call Print_Results

    End If     '530



540:   'Resource Constrained Solution(RCS)
    'Relaxes End Date Constraints and Incorporates Delays in the Schedules with Lags

541:   'Re-Initialize Temporary Variables
    For i = 1 To NP
      For j = 1 To JT
        TES(i, j) = ES(i, j)
        TEF(i, j) = EF(i, j)
        TLS(i, j) = LS(i, j)
        TLF(i, j) = LF(i, j)
        TTS(i, j) = TS(i, j)
      Next
    Next
```

```
     'Select Project Priority Order, First Time Clears Assignment accross all projects
542:   For n = 1 To NP
        If PP = 0 Then          'PP=0 for Sequentially ordered projects
          i = n
        End If
        If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
        End If
        If PP = 3 Then          'PP=3 for Random Project Order
          i = RPRI(n)
        End If

        'Check to see if all tasks are assigned
543:      For j = 2 To JT - 1 'Check all tasks to see if they are assigned

          'If any unassigned tasks A(i,j)=0 reinitialize project
544:        If AMJ(i, j) = 0 Then

            'Identifies projects that are not completely scheduled
            RS(i) = 1

            'Re-Initialize AMJ
546:          For jj = 1 To JT - 1
              AMJ(i, jj) = 0 'reset AMJ on subsequent tasks current project
            Next


            'Re-Initialize AJT
548:          For jj = 1 To JT - 1
              For TT = 1 To TP
                AJT(i, jj, TT) = 0 'reset AJT on subsequent tasks current project
              Next
            Next

            'Re-Initialize ST and WS
550:          For jj = 1 To JT
              'TES(i, jj) = ES(i, jj)
              'TEF(i, jj) = EF(i, jj)
              ST(i, jj) = 0
              FN(i, jj) = 0
              WS(i, jj) = TS(i, jj)
              SC(i, jj) = 0
              LG(i, jj) = 0
            Next
          End If    '544
        Next        '543 Check all tasks on a project to see if they are assigned
      Next          '542 Check All projects


      'Select Priority Order for Projects, Completes reassignment one project at a time
562:  For n = 1 To NP
        If PP = 0 Then          'PP=0 for Sequentially ordered projects
          i = n
        End If
        If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
        End If
        If PP = 3 Then          'PP=3 for Random Project Order
          i = RPRI(n)
        End If


564:  If RS(i) = 1 Then 'Reschedule an entire project

        'Assign tasks one at a time
566:      For j = 2 To JT - 1

          'Initialize ST
           ST(i, j) = TES(i, j) + 1

          'Assign Resources one at a time
568:        For m = 1 To MR  'Assign resource sequentially one at a time

            'Checks for Resource Type match
570:          If FJ(i, j) = GM(m, MR) Then

              'Checks if resource m is avail during period
              MA(i, j) = 0
              For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                For ii = 1 To NP
                  For jj = 1 To JT
                    If AJT(ii, jj, t) = m Then
                      MA(i, j) = 1 'set to 1 if not avail
                    End If
                  Next
```

```
                     Next
                  Next

                  'If M equals 0 then m is available, assign to task j during required interval
572:                If MA(i, j) = 0 Then

                     AMJ(i, j) = m 'Assign resource m to proj i task j
                     FN(i, j) = ST(i, j) + DM(i, j) - 1 'Update Finish Date

                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        AJT(i, j, t) = m 'assign m to task k in period t
                     Next
                     GoTo Delayedtask 'Redirect to next task
                  End If
               End If 'FJ=GM
            Next 'Try to assign all resources directly until all have been tested


            'If not assigned, use slack if available to delay task
574:           Do While MA(i, j) = 1 And SC(i, j) < TS(i, j) 'Use slack if it exists

              'Initialize
               SC(i, j) = SC(i, j) + 1
               WS(i, j) = TS(i, j) - SC(i, j)
               ST(i, j) = TES(i, j) + SC(i, j)

576:                For m = 1 To MR

578:                If FJ(i, j) = GM(m, MR) Then ' this checks for skill level match

                     MA(i, j) = 0
                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1 'Checks if resource is avail
during period
                        For iii = 1 To NP
                           For jjj = 1 To JT
                              If AJT(iii, jjj, t) = m Then ' if not avail assign MA = 1
                                 MA(i, j) = 1 'set to 1 if resource is not avail in required period
                              End If

                           Next
                        Next
                     Next

579:                   If MA(i, j) = 0 Then 'If m is available in required interval, assign to j

                        AMJ(i, j) = m 'Assign resource m to proj i task j
```

```
                    FN(i, j) = ST(i, j) + DM(i, j) - 1

                    For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                       AJT(i, j, t) = m 'assign m to task k in period t
                    Next

                    'Forward Pass to Update ES and EF
                    MAXEFPRC(i) = 0
                    For jj = j To JT 'start with task 2, search all rows in PRC for indicator
                       TES(i, jj) = ST(i, jj) - 1  ""-1?
                       TEF(i, jj) = TES(i, jj) + DM(i, jj)
                       For jr = 1 To JT
                          If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                             MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                          End If
                          If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                             TES(i, jj) = TEF(i, jr)
                             TEF(i, jj) = TES(i, jj) + DM(i, jj)
                          End If
                       Next
                    Next
                    GoTo Delayedtask 'Redirect to next task
                 End If
              End If 'FJ=DM
           Next 'try resources at each level of slack
        Loop  'End do while loop to utilize slack


        'If still not assigned increment lag until a resource is free
580:       Do While AMJ(i, j) = 0

           'Initialize
           LG(i, j) = LG(i, j) + 1
           ST(i, j) = TES(i, j) + SC(i, j) + LG(i, j)


           'Assign resource sequentially one at a time
582:          For m = 1 To MR

              'Check for Resource Type match
584:             If FJ(i, j) = GM(m, MR) Then

                 'Checks if resource m is avail during period
                 MA(i, j) = 0
                 For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    For ii = 1 To NP
```

```
                                    For jj = 1 To JT
                                        If AJT(ii, jj, t) = m Then
                                            MA(i, j) = 1 'set to 1 if not avail
                                        End If

                                    Next
                                Next
                            Next

                            'If M still 0 then m is available assign resource m to task j
585:                          If MA(i, j) = 0 Then

                                AMJ(i, j) = m 'Assign resource m to proj i task j
                                'ES(i, j) = ST(i, j)  'Increment Current Task
                                FN(i, j) = ST(i, j) + DM(i, j) - 1
                                For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                                    AJT(i, j, t) = m 'assign m to task k in period t
                                Next

                              'Forward Pass to Update TES and TEF
                                MAXEFPRC(i) = 0
586:                               For jj = j To JT
                                    TES(i, jj) = ST(i, jj) - 1
                                    TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                    For jr = 1 To JT
                                        If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                            MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                                        End If
                                        If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                                            TES(i, jj) = TEF(i, jr)
                                            TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                        End If
                                    Next
                                Next

                                GoTo Delayedtask 'Redirect to next task

                            End If  '585
                        End If      '584 FJ(i,j)not equal GM(i,j) go to next resource
                    Next            '582 Iterate next resource
                Loop                '580 End lag loop
Delayedtask:
            Next                    '566 Iterate next task
        End If                      '564 RSi = 1
    Next                            '562 Iterate next project
    'Last node has TES/TEF equal to largest of TEF of predecessors
588: For i = 1 To NP
        MAXTEFPRC(i) = 0
         For jr = 1 To JT
             If PRC(i, jr, JT) = 1 And TEF(i, jr) >= MAXTEFPRC(i) Then
                MAXTEFPRC(i) = TEF(i, jr) 'set it to the max constant
             End If
         Next
     Next

    'Complete Sequential Scheduling Method - Resource Constrained Solution
590:  For i = 1 To NP
        ST(i, JT) = MAXTEFPRC(i)      'Last node start equals SES (updated)
        FN(i, JT) = MAXTEFPRC(i)      'Last node finish equals SEF (updated)

        'If not resourced, set schedule and value to null
        For j = 2 To JT - 1
           If AMJ(i, j) = 0 Then
              ST(i, JT) = 0
              FN(i, JT) = 0
              BF(i) = 0
              DE(i) = 0
           End If

        Next

        'Otherwise complete the schedule
        CP(i) = FN(i, JT)
        BF(i) = SS(i) + BB(i)
        TE(i) = CP(i) + BF(i)
        DE(i) = TE(i) - TL(i)
        If DE(i) < 0 Then
           DE(i) = 0
        End If
     Next


     'Compute VE and NPVE for the projects that are scheduled
591:  For i = 1 To NP
        VE(i) = 0
        NPVE(i) = 0
        If TE(i) > 0 Then
           VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))
```

```
                'Compute Expected NPV
                GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)

                REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
                GER(i) = REK(i) / (LP(i) / 2)
                DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
                DFPFLE(i) = (1 + IR) ^ -TE(i)
                DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
                DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 + IR) ^ (LP(i) / 2))))
                DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))

                NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) - (GER(i) * DFPGRE(i))) * DFPFRE(i)

            End If
        Next

    'Compute Total Net Present Value Expected TNPVE
    SUMNPVE = 1
    For ii = 1 To NP
        SUMNPVE = SUMNPVE + NPVE(ii)
    Next

    'Compute Program Schedule Efficiency
    PSE = SUMNPVE / SUMNPVP

    SSMRCS(MR) = PSE


        'Format Variables for printing, this preserves the original solution
592:    If MMR = 0 Then     'Print only if a single resource set is run
        PSEO = PSE
        SUMNPVEO = SUMNPVE
        SUMNPVPO = SUMNPVP

        For i = 1 To NP
            CPO(i) = CP(i)
            BFO(i) = BF(i)
            TEO(i) = TE(i)
            DEO(i) = DE(i)
            NPVEO(i) = NPVE(i)

            For j = 1 To JT
                DMO(i, j) = DM(i, j)
        DLO(i, j) = DL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = ES(i, j)
        EFO(i, j) = EF(i, j)
        LSO(i, j) = LS(i, j)
        LFO(i, j) = LF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


593:  ' Print Results of the Resource Constrained Solution(RCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "500 - SEQUENTIAL
SCHEDULING METHOD (SSM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "RESOURCE
CONSTRAINED SOLUTION (RCS)"
    Row = Row + 2

    Call Print_Results

    End If    '592

      'If this solution is best, save it
       BRCS(MR) = 0   'Since the SSMRCS is first, set to zero as initial best solution

595:      If SSMRCS(MR) > BRCS(MR) Then

        BEST(MR) = 1

        BRCS(MR) = SSMRCS(MR)   'Becomes new best solution
        For i = 1 To NP
          BCP(i) = CP(i)
          BBF(i) = BF(i)
          BTE(i) = TE(i)
          BDE(i) = DE(i)
          BNPVE(i) = NPVE(i)

          For j = 1 To JT
            BDM(i, j) = DM(i, j)
            BDL(i, j) = DL(i, j)
            BFJ(i, j) = FJ(i, j)
            BES(i, j) = ES(i, j)
            BEF(i, j) = EF(i, j)
            BLS(i, j) = TLS(i, j)
            BLF(i, j) = TLF(i, j)
            BST(i, j) = ST(i, j)
            BFN(i, j) = FN(i, j)
            BCT(i, j) = CT(i, j)
            BTS(i, j) = TS(i, j)
            BFS(i, j) = FS(i, j)
            BWS(i, j) = WS(i, j)
            BSC(i, j) = SC(i, j)
            BLG(i, j) = LG(i, j)
            BAMJ(i, j) = AMJ(i, j)
            BAHJ(i, j) = AHJ(i, j)

            For t = 1 To TP
              BAJT(i, j, t) = AJT(i, j, t)
            Next
          Next
        Next
      End If        '595

599:  'End 500-599 Sequential Scheduling Method (SSM)

End Sub

Sub Critical_Scheduling_Method()

600:  'Critical Scheduling Method is a multi-pass assignment algorithm.
    'The first sequence develops CRITICAL TASK SOLUTION (CTS) without
changing EF.
     'The CTS assigns tasks in criticality order 1) critical, 2) near critical and 3) non-
critical
    'Critical and near critical tasks are assigned assuming no slack.
    'Near critical tasks are assigned using slack if available.
    'After 1-3, if a project is not fully resourced then the resources are removed from the
    'lower priority projects in reverse priority order and assigned to higher priority
projects.
    'The last sequence resets projects not fully scheduled, relaxes end dates
    'and sequentially forces an assignment to each task.

601:  ' 'Local Variable Declarations

    Dim CTASK As Integer

    Dim TES(6, 10) As Integer  'Temporary
    Dim TEF(6, 10) As Integer  'Temporary
    Dim TLS(6, 10) As Integer  'Temporary
    Dim TLF(6, 10) As Integer  'Temporary

    'Initialize Local Variables
    For i = 1 To NP
      For j = 1 To JT
        TES(i, j) = ES(i, j)
        TEF(i, j) = EF(i, j)
        TLS(i, j) = LS(i, j)
        TLF(i, j) = LF(i, j)
      Next
    Next


602:  'Initialize Variables
    For i = 1 To NP
      RS(i) = 0
      For j = 1 To JT
        AMJ(i, j) = 0
        ST(i, j) = 0
        FN(i, j) = 0
        WS(i, j) = TS(i, j)
        SC(i, j) = 0
        For t = 1 To TP
          AJT(i, j, t) = 0
        Next
      Next
    Next

    ' First assign CT=1 (critical), then CT=2 (near critical), then CT=3 (non critical)
603:  For CTASK = 1 To 3

604:      If CTASK = 1 Then
```

```
              'Select Priority Order for Projects
605:             For n = 1 To NP
                  If PP = 0 Then                    'PP=0 for Sequentially ordered projects
                      i = n
                  End If
                  If PP = 1 Then                    'PP=1 for PRI Productivity prioritized projects
                      i = PRI(n)
                  End If
                  If PP = 2 Then                    'PP=2 for NPVP Potential NPV prioritized projects
                      i = NPVPPRI(n)
                  End If
                  If PP = 3 Then                     'PP=3 for Random Project Order
                      i = RPRI(n)
                  End If

                  'Assign Critical Tasks sequentially one at a time
606:                 For j = 2 To JT - 1

                      'Initialize
                      ST(i, j) = TES(i, j) + 1

                      'Check iif task is critical
607:                     If CT(i, j) = 1 Then

                          'Assign resources to critical tasks one at a time
608:                         For m = 1 To MR

                              'Check for Resource Type match
609:                             If FJ(i, j) = GM(m, MR) Then

                                 'Resource m must be available during period
                                 MA(i, j) = 0
610:                                For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                                     For ii = 1 To NP
                                         For jj = 1 To JT
                                             If AJT(ii, jj, t) = m Then
                                                 MA(i, j) = 1 'set to 1 if resource is not avail
                                             End If
                                         Next
                                     Next
                                 Next

                                 'If m is available in required interval, assign to j
611:                                If MA(i, j) = 0 Then
                                     AMJ(i, j) = m
```

```
                        FN(i, j) = ST(i, j) + DM(i, j) - 1
                        For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                          AJT(i, j, t) = m 'assign m to task k in period t
                        Next
                        GoTo CT1NEXTTASK 'Redirect to next task
                      End If

                    End If  '609 type match FJ=GM
                  Next      '608 Iterate all m resources
                End If      '607 If CT=1
CT1NEXTTASK:
            Next            '610 Try directly assigning all critical tasks
          Next              '606 All projects
        End If              '605 Only Critical CT=1 Set


        'Near Critical Task Assignment
612:      If CTASK = 2 Then

          'Select Priority Order for Projects
614:        For n = 1 To NP
            If PP = 0 Then       'PP=0 for Sequentially ordered projects
              i = n
            End If
            If PP = 1 Then       'PP=1 for PRI Productivity prioritized projects
              i = PRI(n)
            End If
            If PP = 2 Then       'PP=2 for NPVP Potential NPV prioritized projects
              i = NPVPPRI(n)
            End If
            If PP = 3 Then        'PP=3 for Random Project Order
              i = RPRI(n)
            End If

            'Assign Near Critical Tasks
615:          For j = 2 To JT - 1
              ST(i, j) = TES(i, j) + 1

              'Check that task is NEAR critical
616:            If NC(i, j) = 1 Then

                'Assign resources one at a time
617:              For m = 1 To MR

                  'Check for Resource Type match
618:                If FJ(i, j) = GM(m, MR) Then
```

```
                  'Resource m must be available during period
                  MA(i, j) = 0
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If AJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If
                      Next
                    Next
                  Next

                  'If m is available in required interval, assign to j
                  If MA(i, j) = 0 Then

                    AMJ(i, j) = m
                    FN(i, j) = ST(i, j) + DM(i, j) - 1

                    For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                      AJT(i, j, t) = m 'assign m to task k in period t
                    Next
                    GoTo CT2NEXTTASK 'Redirect to next task
                  End If

                End If  '618 type match FJ=GM
              Next      '617 Try all m resources
            End If      '616 If NC=1
CT2NEXTTASK:
          Next          '615 All near critical tasks
        Next            '614 All projects
      End If            '612 End CT=2 Only Near Critical


    'Non Critical Task Assignment
619:  If CTASK = 3 Then 'Now Assign Non-Critical Task Set

      'Select Priority Order for Projects
620:    For n = 1 To NP
        If PP = 0 Then        'PP=0 for Sequentially ordered projects
          i = n
        End If
        If PP = 1 Then        'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then        'PP=2 for NPVP Potential NPV prioritized projects
```

```
            i = NPVPPRI(n)
          End If
          If PP = 3 Then          'PP=3 for Random Project Order
            i = RPRI(n)
          End If

          'Start Assignment of Non-Critical Tasks
621:        For j = 2 To JT - 1

            'Initialize
            ST(i, j) = TES(i, j) + 1 'Start at period ES +1

            'Check that only non-critical and non-near critical
622:          If CT(i, j) = 0 And NC(i, j) = 0 Then

            'Direct assigment of resources
623:            For m = 1 To MR

                'Check Resource Type match
624:              If FJ(i, j) = GM(m, MR) Then

                  'Resource m must be available
                  MA(i, j) = 0
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If AJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If
                      Next
                    Next
                  Next

                  'If m is available in required interval, assign to j
                  If MA(i, j) = 0 Then

                    AMJ(i, j) = m
                    FN(i, j) = ST(i, j) + DM(i, j) - 1

                    For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                      AJT(i, j, t) = m 'assign m to task k in period t
                    Next
                    GoTo CT3NEXTTASK 'Redirect to next task
                  End If
                End If '624  FJ=GM
              Next      '623 All m
            'For Non-Critical Tasks, delay and use slack if available
630:         Do While SC(i, j) < TS(i, j)

              SC(i, j) = SC(i, j) + 1
              WS(i, j) = TS(i, j) - SC(i, j)
              ST(i, j) = TES(i, j) + SC(i, j)

631:           For m = 1 To MR

632:             If FJ(i, j) = GM(m, MR) Then

                  'Checks if resource is avail during period
                  MA(i, j) = 0
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                     For ii = 1 To NP
                        For jj = 1 To JT
                           If AJT(ii, jj, t) = m Then
                              MA(i, j) = 1 'set to 1 if not avail
                           End If

                        Next
                     Next
                  Next


                  'If m is available in required interval, assign to j
                  If MA(i, j) = 0 Then

                     AMJ(i, j) = m 'Assign resource m to proj i task j
                     FN(i, j) = ST(i, j) + DM(i, j) - 1

                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        AJT(i, j, t) = m 'assign m to task k in period t
                     Next


                     'Forward Pass to Update TES and TEF
                     MAXEFPRC(i) = 0
                     For jj = j To JT
                        TES(i, jj) = ST(i, jj) - 1
                        TEF(i, jj) = TES(i, jj) + DM(i, jj)
                        For jr = 1 To JT
                           If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                              MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
```

```
                                    End If
                                    If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                                       TES(i, jj) = TEF(i, jr)
                                       TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                    End If
                                 Next
                              Next

                              GoTo CT3NEXTTASK 'Redirect to next task

                           End If
                        End If   '632 FJ=GM
                     Next        '631 Try all m resource
                  Loop           '630 Try until slack is used up
               End If            '622
CT3NEXTTASK:
            Next                 '621 All tasks
         Next                    '620 All projects
      End If                     '619 CT=3
   Next                          '603 Iterate for CT=1,2,3

      'Complete Schedule
634:    For i = 1 To NP
         ST(i, JT) = TES(i, JT)
         FN(i, JT) = TEF(i, JT)
         For j = 2 To JT - 1
            If AMJ(i, j) = 0 Then
               ST(i, JT) = 0
               FN(i, JT) = 0
               BF(i) = 0
               DE(i) = 0
            End If
         Next
      Next

635:    For i = 1 To NP
         CP(i) = FN(i, JT)
         TE(i) = CP(i) + BF(i)
         DE(i) = TE(i) - TL(i)
         If DE(i) < 0 Then
            DE(i) = 0
         End If
      Next


      'Compute VE and NPVE for the projects that are scheduled
```

```
636:  For i = 1 To NP
      VE(i) = 0
      NPVE(i) = 0
      If TE(i) > 0 Then
        VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))

        'Compute Expected NPV
        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
        REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1)
/ ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
        DFPFLE(i) = (1 + IR) ^ -TE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
        NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)
      End If
    Next    '636

    'Compute Total Net Present Value Expected TNPVE
637:  SUMNPVE = 1
    For ii = 1 To NP
      SUMNPVE = SUMNPVE + NPVE(ii)
    Next

    'Compute Program Schedule Efficiency
638:  PSE = SUMNPVE / SUMNPVP

    CSMTCS(MR) = PSE


    'Print Results Critical Scheduling Method(CSM) Method
    'Time Constrained Solution (TCS)
    'Format Variables for printing, this preserves the original solution
639:  If MMR = 0 Then    'Print only if a single resource set is run
    PSEO = PSE
    SUMNPVEO = SUMNPVE
    SUMNPVPO = SUMNPVP

    For i = 1 To NP
      CPO(i) = CP(i)
      BFO(i) = BF(i)
      TEO(i) = TE(i)
      DEO(i) = DE(i)
      NPVEO(i) = NPVE(i)

      For j = 1 To JT
        DMO(i, j) = DM(i, j)
        DLO(i, j) = DL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = TES(i, j)
        EFO(i, j) = TEF(i, j)
        LSO(i, j) = LS(i, j)
        LFO(i, j) = LF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


640:  'Print Results of the Critical Task Assignment Methd (CTA-TCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "CRITICAL SCHEDULING
METHOD(CSM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "TIME CONSTRAINED
SOLUTION (TCS)"
    Row = Row + 2

    Call Print_Results

    End If     '639

    'Begin Critical REASSIGNMENT If Projects are not scheduled begin removing
resources
    'from lower priority projects and allocate them to higher priority
    'projects based on project priorities and critical task status
```

```
641:  'Initialize Project Counter, will be decremented as projects become non-critical
      NPC = NP

      'Initialize RS(i)
      For i = 1 To NP
         RS(i) = 0
      Next

      'Select Priority Order for Projects, Identify Projects with vacent assignments
642:  For n = 1 To NPC
         If PP = 0 Then         'PP=0 for Sequentially ordered projects
            i = n
            inpc = n
            ic = n
         End If
         If PP = 1 Then         'PP=1 for PRI Productivity prioritized projects
            i = PRI(n)
            inpc = n
            ic = n
         End If
         If PP = 2 Then         'PP=2 for NPVP Potential NPV prioritized projects
            i = NPVPPRI(n)
            inpc = n
            ic = n
         End If
         If PP = 3 Then         'PP=3 for Random Project Order
            i = RPRI(n)
            inpc = n
            ic = n
         End If

         RSI = 0 'RSI=0 means higher level projects have sufficient resources

         'Test to identify Projects with incomplete assignments
644:       For ic = 1 To NPC
            For j = 2 To JT - 1 'Check all tasks to see if they are assigned

               If AMJ(i, j) = 0 Then  'If unassigned tasks A(i,j)=0 reinitialize
                  RS(i) = 1 '1=Yes reschedule project with insufficient resources
                  RSI = 1   '1=Yes higer level projects have insufficient resources
               End If
            Next
         Next 'End Test

         'CT Reassignment only if a higher level project has insufficient resources
```

```
645:    If RSI = 1 Then

        'Remove resource assignments from lowest priority project, reset variables
646:      inpc = NPC

        'Re-Initialize AMJ
        For jj = 1 To JT - 1        'Remove Assignment on lower priority project
          AMJ(inpc, jj) = 0           'reset AMJ on tasks in lower priority projects
        Next

        'Re-Initialize AJT
        For jj = 1 To JT - 1        'Remove Assignment on lower priority project
          For tnpc = 1 To TP
            AJT(inpc, jj, tnpc) = 0  'reset AJT on tasks in lower priority
          Next
        Next

        'Re-Initialize Variables
        For jj = 1 To JT
          TES(inpc, jj) = ES(inpc, jj)
          TEF(inpc, jj) = EF(inpc, jj)
          ST(inpc, jj) = 0
          FN(inpc, jj) = 0
          WS(inpc, jj) = TS(inpc, jj)
          SC(inpc, jj) = 0
          LG(inpc, jj) = 0
        Next

        'Decrement lowest priority project
        NPC = NPC - 1

        'Assign resources from lowest priority project to higher priority projects
647:      For i = 1 To NPC            'fill assignments on higher priority projects

          'Check if higher priority project is resourced
648:        If RS(i) = 1 Then

            'Check all tasks
649:          For j = 2 To JT - 1
              ST(i, j) = TES(i, j) + 1

              'Try to directly reassign unresourced tasks with no slack
650:            If AMJ(i, j) = 0 And (CT(i, j) = 1 Or NC(i, j) = 1) Then
```

```
               'Try to directly assign a resource if available
651:              For m = 1 To MR  'Assign resource sequentially one at a time


                  'Check for skill level match FJ=GM, check if CT=1 or NC=1
652:                 If FJ(i, j) = GM(m, MR) Then

                     'Checks if resource m is avail during period
                     MA(i, j) = 0
                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        For ii = 1 To NP
                           For jj = 1 To JT
                              If AJT(ii, jj, t) = m Then
                                 MA(i, j) = 1 'set to 1 if not avail
                              End If
                           Next
                        Next
                     Next

                     'If available, assign to task j during required interval
                     If MA(i, j) = 0 Then

                        AMJ(i, j) = m
                        FN(i, j) = ST(i, j) + DM(i, j) - 1 'Update Finish Date

                        For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                           AJT(i, j, t) = m 'assign m to task k in period t
                        Next

                        GoTo CT4NEXTTASK 'Redirect to next task

                     End If
                  End If '652 End FJ=GM
               Next      '651 End all m resources directly
            End If       '650 AMJ=0

            'Try to reassign non-critical tasks with no assignment
654:           If AMJ(i, j) = 0 And (CT(i, j) = 0 Or NC(i, j) = 0) Then

               'For Non-Critical Tasks, delay and use slack if available
655:              Do While SC(i, j) < TS(i, j) 'Use slack if it exists

                  SC(i, j) = SC(i, j) + 1
                  WS(i, j) = TS(i, j) - SC(i, j)
                  ST(i, j) = TES(i, j) + SC(i, j)
```

```
656:                 For m = 1 To MR

657:                   If FJ(i, j) = GM(m, MR) Then

                        'Checks if resource is avail during period
                        MA(i, j) = 0
                        For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                           For ii = 1 To NP
                              For jj = 1 To JT
                                 If AJT(ii, jj, t) = m Then
                                    MA(i, j) = 1 'set to 1 if not avail
                                 End If

                              Next
                           Next
                        Next

                        'If m is available in required interval, assign to j
                        If MA(i, j) = 0 Then

                           AMJ(i, j) = m 'Assign resource m to proj i task j
                           FN(i, j) = ST(i, j) + DM(i, j) - 1

                           For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                              AJT(i, j, t) = m 'assign m to task k in period t
                           Next


                          'Forward Pass to Update TES and TEF
                           MAXEFPRC(i) = 0
                           For jj = j To JT
                              TES(i, jj) = ST(i, jj) - 1
                              TEF(i, jj) = TES(i, jj) + DM(i, jj)
                              For jr = 1 To JT
                                 If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                    MAXEFPRC(i) = TEF(i, jr) 'set it to the max
constant
                                 End If
                                 If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then
                                    TES(i, jj) = TEF(i, jr)
                                    TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                 End If
                              Next
                           Next
                        GoTo CT4NEXTTASK 'Redirect to next task

                      End If
                    End If  '657 End FJ=GM
                   Next      '656 End try next m resource
                Loop         '655 End Try until slack on task is used up
               End If        '654 End Only if AMJ=0
CT4NEXTTASK:

            Next             '649 End Assign all tasks
          End If             '648 End Only if RS(i) = 1
        Next                 '647 End Continue assigning on higer priority projects
      End If                 '645 End Only if RS=1
    Next                     '642 End Iteration through projects


    'Complete Schedule
660:  For i = 1 To NP
      ST(i, JT) = TES(i, JT)
      FN(i, JT) = TEF(i, JT)
      For j = 2 To JT - 1
        If AMJ(i, j) = 0 Then
          ST(i, JT) = 0
          FN(i, JT) = 0
          CP(i) = 0
          BF(i) = 0
          DE(i) = 0
        End If
      Next
    Next

    For i = 1 To NP
      TE(i) = CP(i) + BF(i)
      DE(i) = TE(i) - TL(i)
      If DE(i) < 0 Then
        DE(i) = 0
      End If
    Next

    'Compute VE and NPVE for the projects that are scheduled
661:  For i = 1 To NP
      VE(i) = 0
      NPVE(i) = 0
      If TE(i) > 0 Then

        VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))
        'Compute Expected NPV
        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)

        REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1)
/ ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
        DFPFLE(i) = (1 + IR) ^ -TE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))

        NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)
      End If
    Next   '688

    'Compute Total Net Present Value Expected TNPVE
    SUMNPVE = 1
    For ii = 1 To NP
      SUMNPVE = SUMNPVE + NPVE(ii)
    Next

    'Compute Program Schedule Efficiency
    PSE = SUMNPVE / SUMNPVP

    CSMCCS(MR) = PSE


    'Print Results Critical Scheduling Method (CSM) Method
    'Critical Constrained Solution (CCS)
    'Format Variables for printing, this preserves the original solution
662:  If MMR = 0 Then     'Print only if a single resource set is run
    PSEO = PSE
    SUMNPVEO = SUMNPVE
    SUMNPVPO = SUMNPVP

    For i = 1 To NP
      CPO(i) = CP(i)
      BFO(i) = BF(i)
      TEO(i) = TE(i)
      DEO(i) = DE(i)
      NPVEO(i) = NPVE(i)
      For j = 1 To JT
        DMO(i, j) = DM(i, j)
        DLO(i, j) = DL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = TES(i, j)
        EFO(i, j) = TEF(i, j)
        LSO(i, j) = LS(i, j)
        LFO(i, j) = LF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


663:  'Print Results of the Critical Task Assignment Methd (CSM-CCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "600 - CRITICAL SCHEDULING
METHOD (CSM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "CRITICAL CONSTRAINED
SOLUTION (CCS)"
    Row = Row + 2

    Call Print_Results

    End If     '662

    'BEGIN SEQUENTIAL REASSIGNMENT
    'IF CANNOT SCHEDULE PROJECT USING CRITICAL ASSIGNMENT RULES
    'RESET UNSCHEDULED PROJECTS AND RESCHEDULE USING
SEQUENTIAL
    'TASK ASSIGNMENT RELAX END DATE CONSTRAINTS, INCORPORATES
    'DELAYED SCHEDULE WITH LAGS
```

```
     'Select Priority Order for Projects
664:   For n = 1 To NP
        If PP = 0 Then          'PP=0 for Sequentially ordered projects
          i = n
        End If
        If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
        End If
        If PP = 3 Then          'PP=3 for Random Project Order
          i = RPRI(n)
        End If


665:       For j = 2 To JT - 1 'Check all tasks to see if they are assigned

           'If unassigned tasks A(i,j)=0 reinitialize project
666:         If AMJ(i, j) = 0 Then

              RS(i) = 1 'Identifies projects with insufficient resources

              'Re-Initialize AMJ
              For jj = 1 To JT - 1
                AMJ(i, jj) = 0
              Next

              'Re-Initialize AJT
              For jj = 1 To JT - 1
                For TT = 1 To TP
                  AJT(i, jj, TT) = 0
                Next
              Next

              'Re-Initialize Variables
              For jj = 1 To JT
                TES(i, jj) = ES(i, jj)
                TEF(i, jj) = EF(i, jj)
                ST(i, jj) = 0
                FN(i, jj) = 0
                WS(i, jj) = TS(i, jj)
                SC(i, jj) = 0
                LG(i, jj) = 0
              Next
          End If  '704
        Next      '702
      Next        '700

      'Search projects and reschedule those not having task assignments
      'Select Priority Order for Projects
667:  For n = 1 To NP
        If PP = 0 Then        'PP=0 for Sequentially ordered projects
          i = n
        End If
        If PP = 1 Then        'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then        'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
        End If
        If PP = 3 Then        'PP=3 for Random Project Order
          i = RPRI(n)
        End If


668:      If RS(i) = 1 Then 'Reschedule entire project

          'Schedule tasks one at a time
669:        For j = 2 To JT - 1

          'Initialize ST
          ST(i, j) = TES(i, j) + 1

            'Directly assign a resource if available in required periods
670:          For m = 1 To MR  'Assign resource sequentially one at a time

              'Check for skill type match
671:            If FJ(i, j) = GM(m, MR) Then

                'Checks if resource m is avail during period
                MA(i, j) = 0
                For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                  For ii = 1 To NP
                    For jj = 1 To JT
                      If AJT(ii, jj, t) = m Then
                        MA(i, j) = 1 'set to 1 if not avail
                      End If
                    Next
                  Next
                Next
```

```
                      'If M equals 0 then assign to task j during required interval
                      If MA(i, j) = 0 Then

                         AMJ(i, j) = m 'Assign resource m to proj i task j
                         FN(i, j) = ST(i, j) + DM(i, j) - 1 'Update Finish Date

                         For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                            AJT(i, j, t) = m 'assign m to task k in period t
                         Next
                         GoTo CT5NEXTTASK 'Redirect to next task
                      End If
                   End If     '671 FJ=GM
                Next          '670 Try to assign all resources directly

                'If not assigned, use slack if available to delay task
672:               Do While MA(i, j) = 1 And SC(i, j) < TS(i, j)

                   'Initialize
                   SC(i, j) = SC(i, j) + 1
                   WS(i, j) = TS(i, j) - SC(i, j)
                   ST(i, j) = TES(i, j) + SC(i, j)

673:                 For m = 1 To MR

674:                    If FJ(i, j) = GM(m, MR) Then ' this checks for skill level match

                         'Checks if resource is avail during period
                         MA(i, j) = 0
                         For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                            For iii = 1 To NP
                               For jjj = 1 To JT
                                  If AJT(iii, jjj, t) = m Then
                                     MA(i, j) = 1 'set to 1 if not avail
                                  End If

                               Next
                            Next
                         Next

                         'If m is available in required interval, assign to j
                         If MA(i, j) = 0 Then

                            AMJ(i, j) = m 'Assign resource m to proj i task j
                            FN(i, j) = ST(i, j) + DM(i, j) - 1
```

```
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    AJT(i, j, t) = m 'assign m to task k in period t
                  Next

                  'Forward Pass to Update TES and TEF
                  MAXEFPRC(i) = 0
                  For jj = j To JT
                    TES(i, jj) = ST(i, jj) '- 1
                    TEF(i, jj) = TES(i, jj) + DM(i, jj)
                    For jr = 1 To JT
                       If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                        MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                      End If
                      If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                        TES(i, jj) = TEF(i, jr)
                        TEF(i, jj) = TES(i, jj) + DM(i, jj)
                      End If
                    Next
                  Next

                  GoTo CT5NEXTTASK 'Redirect to next task

                End If

              End If  '674 FJ=DM

            Next      '673 Try resources at each level of slack

        Loop          '672 End do while loop to utilize slack


        'Increment task lag LG by 1 until a resource is free
676:      Do While AMJ(i, j) = 0

          'Initialize
          LG(i, j) = LG(i, j) + 1
          ST(i, j) = TES(i, j) + SC(i, j) + LG(i, j)


678:        For m = 1 To MR  'Assign resource sequentially one at a time

          'Checks for skill type match
680:         If FJ(i, j) = GM(m, MR) Then

              'Checks if resource m is avail during period
              MA(i, j) = 0
```

```
                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        For ii = 1 To NP
                           For jj = 1 To JT
                              If AJT(ii, jj, t) = m Then ' if not avail assign MA = 1
                                 MA(i, j) = 1 'set to 1 if resource is not avail in required period
                              End If

                           Next
                        Next
                     Next

                     'If M still 0 then m is available assign resource m to task j
682:                   If MA(i, j) = 0 Then

                        AMJ(i, j) = m

                        FN(i, j) = ST(i, j) + DM(i, j) - 1
                        For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                           AJT(i, j, t) = m 'assign m to task k in period t
                        Next

                       'Forward Pass to Update TES and TEF
                        MAXEFPRC(i) = 0
                        For jj = j To JT
                           TES(i, jj) = ST(i, jj) - 1
                           TEF(i, jj) = TES(i, jj) + DM(i, jj)
                           For jr = 1 To JT
                              If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                 MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                              End If
                              If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                                 TES(i, jj) = TEF(i, jr)
                                 TEF(i, jj) = TES(i, jj) + DM(i, jj)
                              End If
                           Next
                        Next


                        GoTo CT5NEXTTASK 'Redirect to next task

                     End If      '682
                  End If         '680 if FJ(i,j)not equal GM(i,j) go to next resource
               Next              '678 Iterate to next resource
            Loop                 '676 End Lag loop
CT5NEXTTASK:
         Next                    '669 Loop to next task
```

```
    End If                  '668
  Next                      '667 Iterate to next project


    'Assign Start and End Dates to Last (Artificial) Task
686:  For i = 1 To NP
        ST(i, JT) = TES(i, JT)
        FN(i, JT) = TEF(i, JT)
        CP(i) = FN(i, JT)
    Next

    'Complete schedule
688:  For i = 1 To NP
        BF(i) = SS(i) + BB(i)
        TE(i) = CP(i) + BF(i)
        DE(i) = TE(i) - TL(i)
          If DE(i) < 0 Then
          DE(i) = 0
        End If

    Next

    'Compute VE and NPVE for the projects that are scheduled
689:  For i = 1 To NP
        VE(i) = 0
        NPVE(i) = 0
        If TE(i) > 0 Then

          VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))

          'Compute Expected NPV
          GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
          REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
          GER(i) = REK(i) / (LP(i) / 2)
          DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
          DFPFLE(i) = (1 + IR) ^ -TE(i)
          DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
          DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 + IR) ^ (LP(i) / 2))))
          DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
          NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) - (GER(i) * DFPGRE(i))) * DFPFRE(i)
        End If
    Next
```

```
     'Compute Total Net Present Value Expected TNPVE
     SUMNPVE = 1
690:   For ii = 1 To NP
         SUMNPVE = SUMNPVE + NPVE(ii)
     Next

     'Compute Program Schedule Efficiency
     PSE = SUMNPVE / SUMNPVP
     CSMRCS(MR) = PSE


     'Print Results Critical_Task Scheduling Method (CSM)
     'Time Constrained Solution (RCS)
     'Format Variables for printing, this preserves the original solution
691:   If MMR = 0 Then     'Print only if a single resource set is run
     PSEO = PSE
     SUMNPVEO = SUMNPVE
     SUMNPVPO = SUMNPVP

     For i = 1 To NP
        CPO(i) = CP(i)
        BFO(i) = BF(i)
        TEO(i) = TE(i)
        DEO(i) = DE(i)
        NPVEO(i) = NPVE(i)

        For j = 1 To JT
           DMO(i, j) = DM(i, j)
           DLO(i, j) = DL(i, j)
           FJO(i, j) = FJ(i, j)
           ESO(i, j) = TES(i, j)
           EFO(i, j) = TEF(i, j)
           LSO(i, j) = TLS(i, j)
           LFO(i, j) = TLF(i, j)
           STO(i, j) = ST(i, j)
           FNO(i, j) = FN(i, j)
           CTO(i, j) = CT(i, j)
           TSO(i, j) = TS(i, j)
           FSO(i, j) = FS(i, j)
           WSO(i, j) = WS(i, j)
           SCO(i, j) = SC(i, j)
           LGO(i, j) = LG(i, j)
           AMJO(i, j) = AMJ(i, j)
           AHJO(i, j) = AHJ(i, j)

           For t = 1 To TP
```

```
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


692:  'Print Results
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "600 - CRITICAL SCHEDULING
METHOD"
    Worksheets("Results").Cells(Row + 1, Col).Value = "RESOURCE
CONSTRAINED SOLUTION (RCS)"
    Row = Row + 2

    Call Print_Results

    End If    '691


     'Determine if this solution is best and save it
695:      If CSMRCS(MR) > BRCS(MR) Then

        BEST(MR) = 3

        BRCS(MR) = CSMRCS(MR)   'Becomes new best solution

        For i = 1 To NP
          BCP(i) = CP(i)
          BBF(i) = BF(i)
          BTE(i) = TE(i)
          BDE(i) = DE(i)
          BNPVE(i) = NPVE(i)

          For j = 1 To JT
            BDM(i, j) = DM(i, j)
            BDL(i, j) = DL(i, j)
            BFJ(i, j) = FJ(i, j)
            BES(i, j) = TES(i, j)
            BEF(i, j) = TEF(i, j)
            BLS(i, j) = TLS(i, j)
            BLF(i, j) = TLF(i, j)
            BST(i, j) = ST(i, j)
            BFN(i, j) = FN(i, j)
            BCT(i, j) = CT(i, j)
            BTS(i, j) = TS(i, j)
```

```
                BFS(i, j) = FS(i, j)
                BWS(i, j) = WS(i, j)
                BSC(i, j) = SC(i, j)
                BLG(i, j) = LG(i, j)
                BAMJ(i, j) = AMJ(i, j)
                BAHJ(i, j) = AHJ(i, j)

                For t = 1 To TP
                   BAJT(i, j, t) = AJT(i, j, t)
                Next
             Next
          Next
       End If      '695


699:  'End 600-699 Critical Scheduling Method

End Sub
Sub Early_Scheduling_Method()


700:  'Early Task Assignment is a multi-pass  algorithm that
    'assigns and schedules multiple projects according to a
    'prioritized list of Early Start (ES) dates.
    'After the first pass any project not fully scheduled
    'is rescheduled sequentially.

701:  'Local Variable Declarations
    Dim TES(6, 10) As Integer  'Temp Early Start
    Dim TEF(6, 10) As Integer  'Temp Early Finish
    Dim TLS(6, 10) As Integer  'Temp Late Start
    Dim TLF(6, 10) As Integer  'Temp Late Finish
    Dim TTS(6, 10) As Integer  'Temp Total Slack

    'Initialize Local Variables
    For i = 1 To NP
       For j = 1 To JT
          TES(i, j) = ES(i, j)
          TEF(i, j) = EF(i, j)
          TLS(i, j) = LS(i, j)
          TLF(i, j) = LF(i, j)
          TTS(i, j) = TS(i, j)

       Next
    Next
   'Initialize Variable and Assignments
```

```
      For i = 1 To NP
         RS(i) = 0
         For j = 1 To JT
            AMJ(i, j) = 0
            ST(i, j) = 0
            FN(i, j) = 0
            WS(i, j) = TTS(i, j)
            SC(i, j) = 0

            For t = 1 To TP
               AJT(i, j, t) = 0
            Next
         Next
      Next
      l = 0
         'Tasks are selected in order of Earliest Start Date
710:       For k = 1 To NJT

            For i = 1 To NP
               For j = 1 To JT
                  l = l + 1
                     If ESARO(k) = ESA(l) Then
                        ST(i, j) = ES(i, j) + 1
                        l = 0
                        GoTo STARTES
                     End If
               Next
            Next

STARTES:
            'Try to assign resources sequentially one at a time
712:           For m = 1 To MR

               'Skill level must match in order to consider assignment
713:              If FJ(i, j) = GM(m, MR) Then

                  'Resource m must be available during period
                  MA(i, j) = 0
                  For t = ST(i, j) To ST(i, j) + DM(i, j) - 1

                     For ii = 1 To NP

                        For jj = 1 To JT
                           If AJT(ii, jj, t) = m Then
                              MA(i, j) = 1 'set to 1 if not avail
                           End If
                        Next
                     Next
                  Next

                  'If m is available in required interval, assign to j
                  If MA(i, j) = 0 Then

                     AMJ(i, j) = m 'Assign resource m to proj i task j
                     FN(i, j) = ST(i, j) + DM(i, j) - 1

                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        AJT(i, j, t) = m 'assign m to task k in period t
                     Next
                     GoTo ES1NEXTTASK 'Redirect to next task
                  End If
               End If      '713 FJ=GM
            Next           '712 Iterate all resources directly
                        'No slack if used in the first pass
ES1NEXTTASK:
   Next                    '710 Iterate NJT tasks


      ' Complete schedule
714:     For i = 1 To NP
         ST(i, JT) = TES(i, JT)
         FN(i, JT) = TEF(i, JT)
         BF(i) = SS(i) + BB(i)
         For j = 2 To JT - 1
            If AMJ(i, j) = 0 Then
               ST(i, JT) = 0
               FN(i, JT) = 0
               BF(i) = 0
               DE(i) = 0
            End If
         Next
      Next
      For i = 1 To NP

         CP(i) = FN(i, JT)
         TE(i) = CP(i) + BF(i)
         DE(i) = TE(i) - TL(i)
         If DE(i) < 0 Then
            DE(i) = 0
         End If
      Next
      'Compute VE and NPVE for the projects that are scheduled
715:   For i = 1 To NP
        VE(i) = 0
        NPVE(i) = 0
        If TE(i) > 0 Then
           VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))

           'Compute Expected NPV
           GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
           REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
           GER(i) = REK(i) / (LP(i) / 2)
           DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) - 1)
/ ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
           DFPFLE(i) = (1 + IR) ^ -TE(i)
           DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
           DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
           DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
           NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

        End If
      Next

      'Compute Total Net Present Value Expected TNPVE
      SUMNPVE = 1
716:   For ii = 1 To NP
        SUMNPVE = SUMNPVE + NPVE(ii)
      Next

      'Compute Program Schedule Efficiency
717:   PSE = SUMNPVE / SUMNPVP

      ESMTCS(MR) = PSE


      'Print results of Early_Start Scheduling Method Initial Pass
      'Time Constrained Solution (TCS)
      'Format Variables for printing, this preserves the original solution
718:   If MMR = 0 Then     'Print only if a single resource set is run
      PSEO = PSE

      For i = 1 To NP
        CPO(i) = CP(i)
        BFO(i) = BF(i)
      TEO(i) = TE(i)
      DEO(i) = DE(i)
      NPVEO(i) = NPVE(i)

      For j = 1 To JT
        DMO(i, j) = DM(i, j)
        DLO(i, j) = DL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = TES(i, j)
        EFO(i, j) = TEF(i, j)
        LSO(i, j) = LS(i, j)
        LFO(i, j) = LF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


719:  'Print Results of the Early_Start Scheduling Methd (ESM-TCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "700 - EARLY SCHEDULING
METHOD (ESM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "TIME CONSTRAINED
SOLUTION (TCS)"
    Row = Row + 2

    Call Print_Results

    End If    '718


    'Second Pass, Sequential Project Scheduling
    'Identify any projects with incomplete assignments
```

```
740:  For n = 1 To NP
       If PP = 0 Then          'PP=0 for Sequentially ordered projects
          i = n
       End If
       If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
       End If
       If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
       End If
       If PP = 3 Then           'PP=3 for Random Project Order
          i = RPRI(n)
       End If

       RSI = 0 'Initialize RSI=0 means projects have sufficient resources

       'Identify Projects with vacent assignments, reset variables
741:      For j = 2 To JT - 1 'Check all tasks to see if they are assigned

        'If unassigned tasks A(i,j)=0 reinitialize project
742:         If AMJ(i, j) = 0 Then

             RS(i) = 1 '1=Yes reschedule project with insufficient resources

             For jj = 1 To JT - 1
                AMJ(i, jj) = 0
                WS(i, jj) = TTS(i, jj)
                SC(i, jj) = 0
                ST(i, jj) = 0
                FN(i, jj) = 0
                LG(i, jj) = 0

                For TT = 1 To TP
                   AJT(i, jj, TT) = 0
                Next
             Next
          End If    '742
       Next         '741
    Next            '740


       'Execute Sequential Assignment on Projects RS(i)=1
745:      For n = 1 To NP
          If PP = 0 Then          'PP=0 for Sequentially ordered projects
             i = n
          End If
        If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
          i = PRI(n)
        End If
        If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
          i = NPVPPRI(n)
        End If
        If PP = 3 Then          'PP=3 for Random Project Order
          i = RPRI(n)
        End If

        'Reschedule entire project
746:      If RS(i) = 1 Then

          'Reschedule one task at a time
748:        For j = 2 To JT - 1

            'Initialize ST
            ST(i, j) = TES(i, j) + 1

            'Directly assign a resource if available in required periods
750:          For m = 1 To MR  'Assign resource sequentially one at a time

              'Check for skill type match
752:            If FJ(i, j) = GM(m, MR) Then

                'Checks if resource m is avail during period
                MA(i, j) = 0
                For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                  For ii = 1 To NP
                    For jj = 1 To JT
                      If AJT(ii, jj, t) = m Then
                        MA(i, j) = 1 'set to 1 if not avail
                      End If
                    Next
                  Next
                Next

                'If M equals 0 then m is available, assign to task j
                If MA(i, j) = 0 Then
                  AMJ(i, j) = m
                  FN(i, j) = ST(i, j) + DM(i, j) - 1 'Update Finish Date
                    For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                      AJT(i, j, t) = m 'assign m to task k in period t
                    Next
                  GoTo ES5NEXTTASK 'Redirect to next task
                End If
```

```
               End If    '752 FJ=GM
            Next         '750 Try to assign all resources directly

            'If not assigned, use slack if available to delay task
754:          Do While MA(i, j) = 1 And SC(i, j) < TTS(i, j)

               'Initialize
               SC(i, j) = SC(i, j) + 1
               WS(i, j) = TTS(i, j) - SC(i, j)
               ST(i, j) = TES(i, j) + SC(i, j)

756:             For m = 1 To MR

                  'Check for skill type match
758:                If FJ(i, j) = GM(m, MR) Then

                     'Checks if resource is avail during period
                     MA(i, j) = 0
                     For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                        For iii = 1 To NP
                           For jjj = 1 To JT
                              If AJT(iii, jjj, t) = m Then
                                 MA(i, j) = 1 'set to 1 if not avail
                              End If
                           Next
                        Next
                     Next

                     'If m is available in required interval, assign to j
                     If MA(i, j) = 0 Then
                        AMJ(i, j) = m 'Assign resource m to proj i task j
                        FN(i, j) = ST(i, j) + DM(i, j) - 1

                        For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                           AJT(i, j, t) = m 'assign m to task k in period t
                        Next


                        'Forward Pass to Update TES and TEF
                        MAXEFPRC(i) = 0
                        For jj = j To JT
                           TES(i, jj) = ST(i, jj) - 1
                           TEF(i, jj) = TES(i, jj) + DM(i, jj)
                           For jr = 1 To JT
                              If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                 MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
```

```
                          End If
                          If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                             TES(i, jj) = TEF(i, jr)
                             TEF(i, jj) = TES(i, jj) + DM(i, jj)
                          End If
                       Next
                    Next
                    GoTo ES5NEXTTASK 'Redirect to next task
                 End If
              End If        '758 FJ=DM
           Next             '756 Try resources at each level of slack
        Loop                '754 Loop while slack is available


        'If still not assigned increment lag until a resource is free
760:       Do While AMJ(i, j) = 0

           'Initialize
           LG(i, j) = LG(i, j) + 1
           ST(i, j) = TES(i, j) + SC(i, j) + LG(i, j)

           'Assign resource sequentially one at a time
762:          For m = 1 To MR

              'Check for resource type match
764:             If FJ(i, j) = GM(m, MR) Then

                 'Checks if resource m is avail during period
                 MA(i, j) = 0
                 For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                    For ii = 1 To NP
                       For jj = 1 To JT
                          If AJT(ii, jj, t) = m Then
                             MA(i, j) = 1 'set to 1 if not avail
                          End If
                       Next
                    Next
                 Next

                 'If M still 0 then m is available assign resource m to task j
766:                If MA(i, j) = 0 Then

                    AMJ(i, j) = m 'Assign resource m to proj i task j
                    FN(i, j) = ST(i, j) + DM(i, j) - 1
                       For t = ST(i, j) To ST(i, j) + DM(i, j) - 1
                          AJT(i, j, t) = m 'assign m to task k in period t
```

```
                               Next

                                'Forward Pass to Update TES and TEF
                                MAXEFPRC(i) = 0
                               For jj = j To JT
                                  TES(i, jj) = ST(i, jj) - 1
                                  TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                  For jr = 1 To JT
                                     If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                        MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                                     End If
                                     If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then

                                        TES(i, jj) = TEF(i, jr)
                                        TEF(i, jj) = TES(i, jj) + DM(i, jj)
                                     End If
                                  Next
                               Next

                               GoTo ES5NEXTTASK 'Redirect to next task
                            End If          '766
                         End If             '764
                      Next                  '762 Increment to next resource
                Loop                        '760 End lag loop
ES5NEXTTASK:
             Next                           '748 Iterate to next task
          End If                            '746 Only reassign projects where RS(i) =1
        Next                                '745 Iterate next project


        'Assign Start and End Dates to Last (Artificial) Task
768:      For i = 1 To NP
           ST(i, JT) = TES(i, JT)
           FN(i, JT) = TEF(i, JT)
        Next

        'Complete schedule
770:       For i = 1 To NP
           BF(i) = SS(i) + BB(i)
           CP(i) = FN(i, JT)
           TE(i) = CP(i) + BF(i)
           DE(i) = TE(i) - TL(i)
           If DE(i) < 0 Then
              DE(i) = 0
           End If
          Next


          'Compute VE and NPVE for the projects that are scheduled
772:          For i = 1 To NP
              VE(i) = 0
              NPVE(i) = 0
              If TE(i) > 0 Then
                 VE(i) = VP(i) * (1 - ((3 * DE(i)) / LP(i)) + ((2 * (DE(i) ^ 2)) / (LP(i) ^ 2)))

                 'Compute Expected NPV
                 GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
                 REK(i) = RPK(i) * (1 - (2 * (DE(i) / LP(i))))
                 GER(i) = REK(i) / (LP(i) / 2)
                 DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - DE(i))) - (IR * ((LP(i) / 2) - DE(i))) -
1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - DE(i))))
                 DFPFLE(i) = (1 + IR) ^ -TE(i)
                 DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
                 DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) *
((1 + IR) ^ (LP(i) / 2))))
                 DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
                 NPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)
              End If
          Next

          'Compute Total Net Present Value Expected TNPVE
          SUMNPVE = 1
774:         For ii = 1 To NP
              SUMNPVE = SUMNPVE + NPVE(ii)
          Next

          'Compute Program Schedule Efficiency
776:         PSE = SUMNPVE / SUMNPVP
          ESMRCS(MR) = PSE


       'Print results of Early_Start Scheduling Method
       'Resource Constrained Solution (RCS)
       'Format Variables for printing, this preserves the original solution
790:   If MMR = 0 Then     'Print only if a single resource set is run
       PSEO = PSE
       SUMNPVEO = SUMNPVE
       SUMNPVPO = SUMNPVP
```

```
    For i = 1 To NP
      CPO(i) = CP(i)
      BFO(i) = BF(i)
      TEO(i) = TE(i)
      DEO(i) = DE(i)
      NPVEO(i) = NPVE(i)

      For j = 1 To JT
        DMO(i, j) = DM(i, j)
        DLO(i, j) = DL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = TES(i, j)
        EFO(i, j) = TEF(i, j)
        LSO(i, j) = TLS(i, j)
        LFO(i, j) = TLF(i, j)
        STO(i, j) = ST(i, j)
        FNO(i, j) = FN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = TS(i, j)
        FSO(i, j) = FS(i, j)
        WSO(i, j) = WS(i, j)
        SCO(i, j) = SC(i, j)
        LGO(i, j) = LG(i, j)
        AMJO(i, j) = AMJ(i, j)
        AHJO(i, j) = AHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = AJT(i, j, t)
        Next
      Next
    Next


791:  'Print Results of the Early Scheduling Methd (ESM-RCS)
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "700 - EARLY SCHEDULING
METHOD (ESM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "RESOURCE
CONSTRAINED SOLUTION (RCS)"
    Row = Row + 2

    Call Print_Results

    End If    '790
```

```
        'Determine if this solution is best and save it
795:     If ESMRCS(MR) > BRCS(MR) Then

          BEST(MR) = 3

          BRCS(MR) = ESMRCS(MR)   'Becomes new best solution

          For i = 1 To NP
            BCP(i) = CP(i)
            BBF(i) = BF(i)
            BTE(i) = TE(i)
            BDE(i) = DE(i)
            BNPVE(i) = NPVE(i)

            For j = 1 To JT
              BDM(i, j) = DM(i, j)
              BDL(i, j) = DL(i, j)
              BFJ(i, j) = FJ(i, j)
              BES(i, j) = TES(i, j)
              BEF(i, j) = TEF(i, j)
              BLS(i, j) = TLS(i, j)
              BLF(i, j) = TLF(i, j)
              BST(i, j) = ST(i, j)
              BFN(i, j) = FN(i, j)
              BCT(i, j) = CT(i, j)
              BTS(i, j) = TS(i, j)
              BFS(i, j) = FS(i, j)
              BWS(i, j) = WS(i, j)
              BSC(i, j) = SC(i, j)
              BLG(i, j) = LG(i, j)
              BAMJ(i, j) = AMJ(i, j)
              BAHJ(i, j) = AHJ(i, j)

              For t = 1 To TP
                BAJT(i, j, t) = AJT(i, j, t)
              Next
            Next
          Next
        End If        '795

799:    'End 700-799 Early Scheduling Method

End Sub

Sub Level_Scheduling_Method()
```

```
800:   'LEVEL SCHEDULING METHOD(LSM)
       'All Constructive Schedules developed to this point assume that
       'all resources are Level 2 thus using mean duration to develop
       'the robust schedule.

       'The Level Scheduling Method applies actual Skill Levels already
       'assigned to each task and thus the duration of tasks are modified
       'to reflect different work rate parameters of Q1 and Q3.
       'An updated Critical Path/Critical Chain pass robust schedule is
       'derived using the adjusted duration estimates.
       'Once the durations are modified on a task that task essentially
       'requires two modes for assignment; 1) Resource Type and 2) Skill level.
       'Updated assignments/schedules result in a Level Constrained Solution (LCS).


801:   'Declare Local Variables

        Dim TES(6, 10) As Integer  'Temp Early Start
        Dim TEF(6, 10) As Integer  'Temp Early Finish
        Dim TLS(6, 10) As Integer  'Temp Late Start
        Dim TLF(6, 10) As Integer  'Temp Late Finish
        Dim TTS(6, 10) As Integer  'Temp Total Slack


802:   'Initialize Variables
       SPSE = 0

       For i = 1 To NP
          SCP(i) = 0
          SBF(i) = 0
          SBB(i) = 0
          SSS(i) = 0
          STE(i) = 0
          SDE(i) = 0
          SNPVE(i) = 0
          SVE(i) = 0

          For j = 1 To JT
             TES(i, j) = 0
             TEF(i, j) = 0
             TLS(i, j) = 0
             TLF(i, j) = 0
             TTS(i, j) = 0

             SDM(i, j) = DM(i, j)
```

```
                SDL(i, j) = DL(i, j)
                SES(i, j) = 0
                SEF(i, j) = 0
                SLS(i, j) = 0
                SLF(i, j) = 0
                SST(i, j) = 0
                SFN(i, j) = 0
                STS(i, j) = 0
                SWS(i, j) = 0
                SSC(i, j) = 0
                SLG(i, j) = 0
                SAMJ(i, j) = 0
                SAHJ(i, j) = 0

                For t = 1 To TP
                    SAJT(i, j, t) = 0
                Next
            Next
        Next

        ''' For each task compute a new SDM and SDL if resource is a 1 or 3
805:       For i = 1 To NP

            For j = 2 To JT - 1

                m = BAMJ(i, j)
                h = HM(m, MR)          'skill level for task i,j

                If h = 1 Then
                    SDM(i, j) = (1 - Q1) * SDM(i, j)
                    SDL(i, j) = (1 - Q1) * SDL(i, j)
                    SAHJ(i, j) = h
                End If

                If h = 2 Then
                    SAHJ(i, j) = h
                End If

                If h = 3 Then
                    SDM(i, j) = (1 + Q3) * SDM(i, j)
                    SDL(i, j) = (1 + Q3) * SDL(i, j)
                    SAHJ(i, j) = h
                End If
            Next
        Next
```

```
     'Critical Path Method
     'Forward Pass to Determin SES, SEF
     'Initialize SES(1) and SEF(1)
810: For i = 1 To NP
      SES(i, 1) = 0 + SLG(i, 1)
      SEF(i, 1) = SES(i, 1) + SDM(i, 1)
     Next

     'The earliest start (SES)for any task is equal to the maximum of the
     'earliest finish(SEF)of the immediate predecessors
811: For i = 1 To NP
      For j = 2 To JT 'start with task 2, search all rows in PRC for indicator
         SES(i, j) = 0
         For jr = 1 To JT
            If PRC(i, jr, j) = 1 Then 'it is a predecessor
               MAXEFPRC(i) = SEF(i, jr) 'set it to the max constant
            End If
            If PRC(i, jr, j) = 1 And MAXEFPRC(i) >= SES(i, j) Then
               SES(i, j) = SEF(i, jr) '+ SLG(i, j)
               SEF(i, j) = SES(i, j) + SDM(i, j) + SLG(i, j)
            End If
         Next
      Next
     Next

     'Backward Pass to Determine SLF, SLS
     'Initialize
812: For i = 1 To NP
      SLF(i, JT) = SEF(i, JT)
      SLS(i, JT) = SLF(i, JT) - SDM(i, JT)
     Next

     'The latest finish (SLF) for any task is the smallest
     'of the latest start(SLS) times of immediate successors
813: For i = 1 To NP
      For j = JT - 1 To 1 Step -1
         SLF(i, j) = SLS(i, JT)
         For jr = JT To 1 Step -1
            If PRC(i, j, jr) = 1 Then
               MINLSSUC(i) = SLS(i, jr)
            End If
            If PRC(i, j, jr) = 1 And MINLSSUC(i) <= SLF(i, j) Then
               SLF(i, j) = SLS(i, jr) '- LG(i, jr)
               SLS(i, j) = SLF(i, j) - SDM(i, j)
```

```
              End If
            Next
          Next
        Next


        'Compute New Total Slack
        'Total Slack of activity j is the difference between 'the latest finish SLF(J)
        ' and earliest finish SEF(J),or the difference between SLS(J) and SES(J)
814:    For i = 1 To NP
          For j = 1 To JT
            STS(i, j) = SLF(i, j) - SEF(i, j)
            SWS(i, j) = STS(i, j)
          Next
        Next

        'Initialize temporary ES, EF, LS, LF for updates
815:    For i = 1 To NP
          For j = 1 To JT

            TES(i, j) = SES(i, j)
            TEF(i, j) = SEF(i, j)
            TLS(i, j) = SLS(i, j)
            TLF(i, j) = SLF(i, j)
            TTS(i, j) = STS(i, j)
           Next
        Next


        'Reschedule all projects using new SES, SEF, STS, and SDM
        'Select Priority Order for Projects
824:    For n = 1 To NP
          If PP = 0 Then        'PP=0 for Sequentially ordered projects
            i = n
          End If
          If PP = 1 Then        'PP=1 for PRI Productivity prioritized projects
            i = PRI(n)
          End If
          If PP = 2 Then        'PP=2 for NPVP Potential NPV prioritized projects
            i = NPVPPRI(n)
          End If
          If PP = 3 Then        'PP=3 for Random Project Order
            i = RPRI(n)
          End If
        'Assign tasks one at a time
825:       For j = 2 To JT - 1

            'Initialize ST
            SST(i, j) = TES(i, j) + 1

            'Assign resource sequentially one at a time
826:            For m = 1 To MR

                'Checks for both Resource Type and Skill Level match
827:               If FJ(i, j) = GM(m, MR) And SAHJ(i, j) = HM(m, MR) Then

                    'Checks if resource m is avail during period
                    MA(i, j) = 0
                    For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                        For ii = 1 To NP
                            For jj = 1 To JT
                                If SAJT(ii, jj, t) = m Then
                                    MA(i, j) = 1 'set to 1 if not avail
                                End If
                            Next
                        Next
                    Next

                    'If M equals 0 then assign to task j during required interval
                    If MA(i, j) = 0 Then

                        SAMJ(i, j) = m 'Assign resource m to proj i task j
                        SFN(i, j) = SST(i, j) + SDM(i, j) - 1 'Update Finish Date

                        For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                            SAJT(i, j, t) = m 'assign m to task i,j in period t
                        Next

                        GoTo LSNEXTTASK 'Redirect to next task

                    End If
                End If          '827 FJ=GM
             Next               '826 All resources directly


            'If not assigned, use slack if available to delay task
828:            Do While MA(i, j) = 1 And SSC(i, j) < STS(i, j)

                SSC(i, j) = SSC(i, j) + 1
                SWS(i, j) = TTS(i, j) - SSC(i, j)
            SST(i, j) = TES(i, j) + SSC(i, j)

            'Assign resources one at a time
829:           For m = 1 To MR

                'Check for both Resource Type and Skill Level match
830:               If FJ(i, j) = GM(m, MR) And SAHJ(i, j) = HM(m, MR) Then

                    'Check if resource is avail during period
                    MA(i, j) = 0
                    For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                        For iii = 1 To NP
                            For jjj = 1 To JT
                                If SAJT(iii, jjj, t) = m Then
                                    MA(i, j) = 1 'set to 1 if not avail
                                End If

                            Next
                        Next
                    Next


                    'If m is available in required interval, assign to j
831:                   If MA(i, j) = 0 Then

                        SAMJ(i, j) = m 'Assign resource m to proj i task j
                        SFN(i, j) = SST(i, j) + SDM(i, j)

                        For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                            SAJT(i, j, t) = m 'assign m to task k in period t
                        Next


                        'Forward Pass to Update TES and TEF
                        MAXEFPRC(i) = 0
832:                       For jj = j To JT
                            TES(i, jj) = SST(i, jj) - 1
                            TEF(i, jj) = TES(i, jj) + SDM(i, jj)
                            For jr = 1 To JT
                                If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                                    MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                                End If
                                If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                                    TES(i, jj) = TEF(i, jr)
                                    TEF(i, jj) = TES(i, jj) + SDM(i, jj)
                                End If
```

```
                  Next
                Next

                GoTo LSNEXTTASK 'Redirect to next task
              End If  '831
            End If    '830  FJ=DM
          Next        '829 All resources at each level of slack
        Loop          '828 While slack is available


        'If still not assigned increment lag until a resource is free
835:      Do While SAMJ(i, j) = 0 '

          'Initialize
          SLG(i, j) = SLG(i, j) + 1
          SST(i, j) = TES(i, j) + SSC(i, j) + SLG(i, j)

          'Assign resource sequentially one at a time
836:        For m = 1 To MR

            'Check for both Resource Type and Skill Level match
837:          If FJ(i, j) = GM(m, MR) And SAHJ(i, j) = HM(m, MR) Then


              'Checks if resource m is avail during period
              MA(i, j) = 0
              For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                For ii = 1 To NP
                  For jj = 1 To JT
                    If SAJT(ii, jj, t) = m Then
                      MA(i, j) = 1 'set to 1 if not avail
                    End If
                  Next
                Next
              Next

              'If M still 0 then m is available assign resource m to task j
838:            If MA(i, j) = 0 Then

                SAMJ(i, j) = m 'Assign resource m to proj i task j
                SFN(i, j) = SST(i, j) + SDM(i, j)
                For t = SST(i, j) To SST(i, j) + SDM(i, j) - 1
                  SAJT(i, j, t) = m 'assign m to task k in period t
                Next

                'Forward Pass to Update TES and TEF
                MAXEFPRC(i) = 0
839:              For jj = j To JT
                   TES(i, jj) = SST(i, jj) - 1
                   TEF(i, jj) = TES(i, jj) + SDM(i, jj)
                   For jr = 1 To JT
                     If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                       MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                     End If
                     If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj) Then
                       TES(i, jj) = TEF(i, jr)
                       TEF(i, jj) = TES(i, jj) + SDM(i, jj)
                     End If
                   Next
                Next

                GoTo LSNEXTTASK 'Redirect to next task

              End If      '838
            End If        '837 FJ = GM(i,j)
          Next            '836 Iterate next resource
        Loop              '835 While lag increases
LSNEXTTASK:
      Next                '825 Next task
    Next                  '824 Next project


    'Compute Square Root of Sum of Squares
840:  For i = 1 To NP
      For j = 1 To JT
        SSE(i, j) = (SDL(i, j) - SDM(i, j)) ^ 2
      Next
    Next

    For i = 1 To NP
      SV(i) = 0
    Next

    For i = 1 To NP
      For j = 1 To JT
        SV(i) = SV(i) + (SSE(i, j))
        SSS(i) = Sqr(SV(i))
      Next
    Next

    'Last node has TES/TEF equal to largest of TEF of predecessors
    For i = 1 To NP
```

```
          MAXTEFPRC(i) = 0
           For jr = 1 To JT
               If PRC(i, jr, JT) = 1 And TEF(i, jr) >= MAXTEFPRC(i) Then 'it is a
predecessor
                   MAXTEFPRC(i) = TEF(i, jr) 'set it to the max constant
               End If
           Next
         Next

        'Complete Start and Finish of last node
842:    For i = 1 To NP
          SST(i, JT) = MAXTEFPRC(i)      'Last node start equals SES (updated)
          SFN(i, JT) = MAXTEFPRC(i)      'Last node finish equals SEF (updated)

          'For j = 2 To JT - 1
            'If SAMJ(i, j) = 0 Then   'If not completely resourced, set schedule to null
              'SST(i, JT) = 0      'so that no value is attributed to that project
              'SFN(i, JT) = 0
              'SBF(i) = 0
              'SDE(i) = 0
            'End If
          'Next
          SCP(i) = SFN(i, JT)          'Otherwise complete the schedule

        Next


        'Compute Buffer
844:    For i = 1 To NP
          SBB(i) = 0.1 * SCP(i)
          SBF(i) = SSS(i) + SBB(i)
        Next

        'Compute Critical Chain Duration, equals expected duration
846:    For i = 1 To NP

          STE(i) = SCP(i) + SBF(i)
          SDE(i) = STE(i) - TL(i)
          If SDE(i) < 0 Then
            SDE(i) = 0
          End If
        Next

        'Compute VE and NPVE for the projects that are scheduled
848:    For i = 1 To NP
          SVE(i) = 0
```

```
      SNPVE(i) = 0
      If STE(i) > 0 Then

        SVE(i) = VP(i) * (1 - ((3 * SDE(i)) / LP(i)) + ((2 * (SDE(i) ^ 2)) / (LP(i) ^ 2)))

        'Compute Expected NPV
        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
        REK(i) = RPK(i) * (1 - (2 * (SDE(i) / LP(i))))
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - SDE(i))) - (IR * ((LP(i) / 2) - SDE(i))) -
1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - SDE(i))))
        DFPFLE(i) = (1 + IR) ^ -STE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
        SNPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

      End If
    Next

    'Compute Total Net Present Value Expected TNPVE
850:  SUMNPVE = 1
    For ii = 1 To NP
      SUMNPVE = SUMNPVE + SNPVE(ii)
    Next

    'Compute Program Schedule Efficiency
    SPSE = SUMNPVE / SUMNPVP

    LCS(MR) = SPSE


    'PRINT  SKILL-LEVEL CONSTRAINED SOLUTION (SCS)
    'Format Variables for printing, this preserves the original solution
890:  If MMR = 0 Then     'Print only if a single resource set is run

    PSEO = SPSE
    SUMNPVEO = SUMNPVE
    SUMNPVPO = SUMNPVP

    For i = 1 To NP
      CPO(i) = SCP(i)
      BFO(i) = SBF(i)
      TEO(i) = STE(i)
```

```
         DEO(i) = SDE(i)
         NPVEO(i) = SNPVE(i)

         For j = 1 To JT
            DMO(i, j) = SDM(i, j)
            DLO(i, j) = SDL(i, j)
            FJO(i, j) = FJ(i, j)
            ESO(i, j) = SES(i, j)
            EFO(i, j) = SEF(i, j)
            LSO(i, j) = SLS(i, j)
            LFO(i, j) = SLF(i, j)
            STO(i, j) = SST(i, j)
            FNO(i, j) = SFN(i, j)
            CTO(i, j) = CT(i, j)
            TSO(i, j) = STS(i, j)
            FSO(i, j) = SFS(i, j)
            WSO(i, j) = SWS(i, j)
            SCO(i, j) = SSC(i, j)
            LGO(i, j) = SLG(i, j)
            AMJO(i, j) = SAMJ(i, j)
            AHJO(i, j) = SAHJ(i, j)

            For t = 1 To TP
               AJTO(i, j, t) = SAJT(i, j, t)
            Next
         Next
      Next


891:   'Print Results of the Level Scheduling Method
      Row = Resultsrow
      Col = Excelcol
      Worksheets("Results").Cells(Row, Col).Value = "800 - LEVEL SCHEDULING
METHOD (LSM)"
      Worksheets("Results").Cells(Row + 1, Col).Value = "LEVEL CONSTRAINED
SOLUTION (LCS) RESULTS"
      Row = Row + 2

      Call Print_Results

      End If     '890

899:   'End 800-899 Early Scheduling Method

End Sub
```

```
Sub Level12_Assignment_Method()

900:  'SKILL-LEVEL ASSIGNMENT METHOD (SLAM)
      'This procedure iteratively completes trial solutions of schedules
      'where resources of same type but differing levels are interchanged.
      'This procedures specifically interchanges Level 1 and Level 2.
      'The procedure identifies pairings, exchanges resources including work
      'rates and then reassigns all other resources holding the pairing fixed.
      'Trial solution Total NPV is compared with the best SLAM solution and
      'updated if the results are improved.
      'This procedure is intitialized with the results of the solutions
      'from procedure(800) Level Constrained Solution.


901   'Declare Local Variables

      Dim ix As Integer          'Index of projects
      Dim jx As Integer          'Index of tasks
      Dim iy As Integer          'Index of projects
      Dim jy As Integer          'Index of tasks
      Dim nz As Integer           'Index of projects
      Dim hx As Integer           'Skill level of x
      Dim hy As Integer           'Skill level of y
      Dim my As Integer            'Resource number of x
      Dim mx As Integer            'Resource number of y
      Dim gy As Integer           'Resource type of x
      Dim gx As Integer           'Resource type of y


      Dim TPSE As Single         'Program Schedule Efficiency
      Dim TSUMNPVE As Single      'Total Sum of Program Expected Net Present Value
      Dim TCP(6) As Integer       'Critical Path
      Dim TBB(6) As Single        'Bias Buffer
      Dim TBF(6) As Single        'Total Buffer
      Dim TCC(6) As Integer       'Critical Chain Duration
      Dim TTE(6) As Integer       'Expected project duration, equivalent to CCD
      Dim TDE(6) As Integer       'Expected Delay between potential launch and expected
duration
      Dim TSS(6) As Integer       'Sum of Square Errors
      Dim TVE(6) As Variant       'Expected Value
      Dim TNPVE(6) As Single       'Expected Net Present Value
      Dim TDM(6, 10) As Integer   'Mean task duration estimates
      Dim TDL(6, 10) As Integer   'Low risk task duration estimates
      Dim TES(6, 10) As Integer   'Earliest Start Time task j
      Dim TEF(6, 10) As Integer   'Earliest Finish Time task j
      Dim TLS(6, 10) As Integer   'Late Start Time task j
```

```
    Dim TLF(6, 10) As Integer   'Late Finish Time task j
    Dim TTS(6, 10) As Integer   'Total slack
    Dim TWS(6, 10) As Integer   'Working slack
    Dim TSC(6, 10) As Integer   'Slack counter
    Dim TLG(6, 10) As Integer   'Temporary lag
    Dim TCT(6, 10) As Integer   'Critical Path Task Indicator 1/0
    Dim TSE(6, 10) As Integer   'Square error
    Dim TST(6, 10) As Integer   'Start Time for task j
    Dim TFN(6, 10) As Integer   'Finish Time for task j
    Dim TAMJ(6, 10) As Integer 'Resource assignment matrix
    Dim TAHJ(6, 10) As Integer 'Skill level assignment matrix
    Dim TAJT(6, 10, 800) As Integer 'Assignment schedule matrix

    Dim TEMPX As Integer        'Resource x Substitution
    Dim TEMPY As Integer        'Resource y Substitution
    Dim TEMPMX As Integer         'Resource x id substitution
    Dim TEMPMY As Integer         'Resource y id substitution
    Dim TEMPHX As Integer        'Resource x Skill Level substitution
    Dim TEMPHY As Integer        'Resource y Skill Level Substitution


902:  'Initialize Solution Variables - Equal to Level Constrained Solution
    UPSE = SPSE
    SLAM12(MR) = LCS(MR)

    For i = 1 To NP
      UCP(i) = SCP(i)
      UBF(i) = SBF(i)
      UBB(i) = SBB(i)
      USS(i) = SSS(i)
      UTE(i) = STE(i)
      UDE(i) = SDE(i)
      UNPVE(i) = SNPVE(i)
      UVE(i) = SVE(i)

      For j = 1 To JT
        UDM(i, j) = SDM(i, j)  'From Skill Level Constrained Solution
        UDL(i, j) = SDL(i, j)  'From Skill Level Constrained Solution
        UES(i, j) = SES(i, j)
        UEF(i, j) = SEF(i, j)
        ULS(i, i) = SLS(i, j)
        ULF(i, j) = SLF(i, j)
        UST(i, j) = SST(i, j)
        UFN(i, j) = SFN(i, j)
        UTS(i, j) = STS(i, j)
        UWS(i, j) = SWS(i, j)
        USC(i, j) = SSC(i, j)
        ULG(i, j) = SLG(i, j)

        UAMJ(i, j) = SAMJ(i, j)
        UAHJ(i, j) = SAHJ(i, j)

        For t = 1 To TP
          UAJT(i, j, t) = SAJT(i, j, t)
        Next

      Next
    Next


903:    'Initialize Temporary Variables for Substitution Procedure

    For i = 1 To NP
      TCP(i) = 0
      TBF(i) = 0
      TBB(i) = 0
      TSS(i) = 0
      TTE(i) = 0
      TDE(i) = 0
      TNPVE(i) = 0
      TVE(i) = 0

      For j = 1 To JT
        TDM(i, j) = UDM(i, j)  'From Skill Level Constrained Solution
        TDL(i, j) = UDL(i, j)  'From Skill Level Constrained Solution
        TES(i, j) = 0
        TEF(i, j) = 0
        TLS(i, i) = 0
        TLF(i, j) = 0
        TST(i, j) = UST(i, j)
        TFN(i, j) = UFN(i, j)
        TTS(i, j) = 0
        TWS(i, j) = 0
        TSC(i, j) = 0
        TLG(i, j) = 0

        TAMJ(i, j) = UAMJ(i, j)
        TAHJ(i, j) = UAHJ(i, j)

        For t = 1 To TP
          TAJT(i, j, t) = UAJT(i, j, t)
          Next
        Next
      Next

   'Counters for substitution
   TRIAL12 = 0
   SUB12 = 0

STARTOVER12:

     'Starting at highest priority project and working to the lower priority projects
     'search through tasks sequentially for a level 1 resource assigned to that task
905:  For n = 1 To NP

      If PP = 0 Then          'PP=0 for Sequentially ordered projects
        ix = n
        iy = n
      End If
      If PP = 1 Then          'PP=1 for PRI Productivity prioritized projects
        ix = PRI(n)
        iy = PRI(n)
      End If
      If PP = 2 Then          'PP=2 for NPVP Potential NPV prioritized projects
        ix = NPVPPRI(n)
        iy = NPVPPRI(n)
      End If
      If PP = 3 Then          'PP=3 for Random Project Order
        ix = RPRI(n)
        iy = RPRI(n)
      End If

      'Check resource from task jx
906:    For jx = 2 To JT - 1
        mx = UAMJ(ix, jx)
        gx = GM(mx, MR)
        hx = HM(mx, MR)

        'Check if resource mx has Skill Level hx
907:      If hx = 1 Then   'if task ix,jx requires a level 1, then continue search

          'Check resources on projects iy
908:        For iy = 1 To NP 'Search all projects for a Level 2

            'Check resources on task jy
909:          For jy = 2 To JT - 1
```

```
                  my = UAMJ(iy, jy) 'identify the resource for the task
                  gy = GM(my, MR)
                  hy = HM(my, MR)

                  'Consider reassignment of x for y if:
                  '1) Resource Type of x matches Resource Type of y
                  '2) Skill Level of x is 1 and Skill level of y is 2
                  '3) Task x and y occur in overlapping periods
                  ' A fourth criteria may be considered later; TDM(x)< TDM(y)

910:               If gx = gy And hy = 2 And (TST(ix, jx) <= TFN(iy, jy) And TFN(ix, jx)
>= TST(iy, jy)) Then

TRIAL12 = TRIAL12 + 1   'Count number of trials

                   'Trial solution alters durations and substitutes level of x and y tasks
                   TDM(ix, jx) = UDM(ix, jx) / (1 - Q1)
                   TDL(ix, jx) = UDL(ix, jx) / (1 - Q1)


                   TDM(iy, jy) = (1 - Q1) * UDM(iy, jy)
                   TDL(iy, jy) = (1 - Q1) * UDL(iy, jy)


912:                 'Forward Pass to Determine ES, EF
                   'Initialize ES(1) and EF(1)
                   For i = 1 To NP
                      TES(i, 1) = 0 + TLG(i, 1)
                      TEF(i, 1) = TDM(i, 1) + TES(i, 1)
                   Next


                   'The earliest start (TES)for any task is equal to the maximum of the
                   'earliest finish(TEF)of the immediate predecessors
913:                 For i = 1 To NP
                      For j = 2 To JT
                         TES(i, j) = 0
                         For jr = 1 To JT
                            If PRC(i, jr, j) = 1 Then 'it is a predecessor
                               MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                            End If
                            If PRC(i, jr, j) = 1 And MAXEFPRC(i) >= TES(i, j) Then
                               TES(i, j) = TEF(i, jr)
                               TEF(i, j) = TES(i, j) + TDM(i, j)
                            End If
                  Next
                Next
              Next


915:            'Backward Pass to Determine TLF, TLS
              'Initialize
              For i = 1 To NP
                TLF(i, JT) = TEF(i, JT)
                TLS(i, JT) = TLF(i, JT) - TDM(i, JT)
              Next

              'The latest finish (TLF) for any task is the smallest
              'of the latest start(TLS) times of immediate successors
              For i = 1 To NP
                For j = JT - 1 To 1 Step -1
                  TLF(i, j) = TLS(i, JT)
                  For jr = JT To 1 Step -1
                    If PRC(i, j, jr) = 1 Then
                      MINLSSUC(i) = TLS(i, jr)
                    End If
                    If PRC(i, j, jr) = 1 And MINLSSUC(i) <= TLF(i, j) Then
                      TLF(i, j) = TLS(i, jr)
                      TLS(i, j) = TLF(i, j) - TDM(i, j)
                    End If
                  Next
                Next
              Next

916:          'Compute New Total Slack
            'Total Slack of activity j is the difference between 'the latest finish TLF(J)
            'and earliest finish EF(J),or the difference between TLS(J) and TES(J)
            For i = 1 To NP
              For j = 1 To JT
                TTS(i, j) = TLF(i, j) - TEF(i, j)
              Next
            Next

920:          'Preassignment Process, Substitute and assign to specific tasks

            'Initialize TAMJ, clear matrix except resource substituton in x and y
            TEMPMX = UAMJ(ix, jx)
            TEMPMY = UAMJ(iy, jy)
            For i = 1 To NP
              For j = 1 To JT
                TAMJ(i, j) = 0
```

```
           Next
         Next
         TAMJ(ix, jx) = TEMPMY
         TAMJ(iy, jy) = TEMPMX

         'Assign TAHJ, retain skill levels for subtitution x and y
         TEMPHX = UAHJ(ix, jx)
         TEMPHY = UAHJ(iy, jy)

         TAHJ(ix, jx) = TEMPHY
         TAHJ(iy, jy) = TEMPHX


         'Pre-assign TAJT for x and y
921:       For i = 1 To NP
            For j = 1 To JT
               For t = 1 To TP
                  TAJT(i, j, t) = 0 'reset TAJT on all tasks
               Next
               If i = ix And j = jx Then
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                     TAJT(i, j, t) = TAMJ(ix, jx)
                  Next

               ElseIf i = iy And j = jy Then
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                     TAJT(i, j, t) = TAMJ(iy, jy)
                  Next
               End If
            Next
         Next


         'Select Priority Order for Projects
922:       For nz = 1 To NP
            If PP = 0 Then        'PP=0 for Sequentially ordered projects
               i = nz
            End If
            If PP = 1 Then        'PP=1 for PRI Productivity prioritized projects
               i = PRI(nz)
            End If
            If PP = 2 Then        'PP=2 for NPVP Potential NPV prioritized projects
               i = NPVPPRI(nz)
            End If
            If PP = 3 Then        'PP=3 for Random Project Order
               i = RPRI(nz)
```

```
              End If

924:            For j = 2 To JT - 1

925:              If i = ix And j = jx Then          'Already assigned resource from y
                    GoTo Delayedtask3
                ElseIf i = iy And j = jy Then   'Already assigned resource from x
                    GoTo Delayedtask3
                Else                            ' All other tasks to be assigned


              'Initialize ST
              TST(i, j) = TES(i, j) + 1

              'Try to directly assign a resource if available in required periods
926:            For m = 1 To MR  'Assign resource sequentially one at a time

                'Check for both Resource Type and Skill Level match
928:              If FJ(i, j) = GM(m, MR) And TAHJ(i, j) = HM(m, MR) Then

                  'Checks if resource m is avail during period
                  MA(i, j) = 0
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If TAJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If
                      Next
                    Next
                  Next

                  'If M equals 0 then assign to task j during required interval
                  If MA(i, j) = 0 Then
                    TAMJ(i, j) = m 'Assign resource m to proj i task j
                    TFN(i, j) = TST(i, j) + TDM(i, j) - 1 'Update Finish Date
                    For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                      TAJT(i, j, t) = m 'assign m to task k in period t
                    Next
                    GoTo Delayedtask3 'Redirect to next task
                  End If
                End If        '928
              Next            '926 Try to assign all resources directly

             'If not assigned, use slack if available to delay task
930:            Do While MA(i, j) = 1 And TSC(i, j) < TTS(i, j) 'Use slack if it exists
```

```
                'Initialize
                TSC(i, j) = TSC(i, j) + 1
                TWS(i, j) = TTS(i, j) - TSC(i, j)
                TST(i, j) = TES(i, j) + TSC(i, j)

                'Assign resources one at a time
931:             For m = 1 To MR

                  'Check for both Resource Type and Skill Level match
932:                If FJ(i, j) = GM(m, MR) And AHJ(i, j) = HM(m, MR) Then
                    MA(i, j) = 0
                      For t = ST(i, j) To ST(i, j) + TDM(i, j) - 1
                        For iii = 1 To NP
                          For jjj = 1 To JT
                            If TAJT(iii, jjj, t) = m Then
                              MA(i, j) = 1 'set to 1 if not avail
                            End If
                          Next
                        Next
                      Next

                      'If m is available in required interval, assign to j
933:                   If MA(i, j) = 0 Then

                        TAMJ(i, j) = m 'Assign resource m to proj i task j
                        TFN(i, j) = TST(i, j) + TDM(i, j)

                        For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                          TAJT(i, j, t) = m 'assign m to task k in period t
                        Next

                        'Forward Pass to Update TES and TEF
                        MAXEFPRC(i) = 0
                        For jj = j To JT
                          TES(i, jj) = TST(i, jj) - 1
                          TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                          For jr = 1 To JT
                            If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                              MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                            End If
                            If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then
                              TES(i, jj) = TEF(i, jr)
                              TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                            End If
                      Next
                    Next

                    GoTo Delayedtask3 'Redirect to next task

                  End If      '933

                End If        '932 FJ=DM

              Next            '931 try all resources at each level of slack

            Loop              '930 End loop to utilize slack


            'If still not assigned increment lag until a resource is free
935:          Do While TAMJ(i, j) = 0

            'Initialize
            TLG(i, j) = TLG(i, j) + 1
            TST(i, j) = TES(i, j) + TSC(i, j) + TLG(i, j)

              'Assign resource sequentially one at a time
936:            For m = 1 To MR

                'Check for both Resource Type and Skill Level match
937:              If FJ(i, j) = GM(m, MR) And TAHJ(i, j) = HM(m, MR) Then

                  'Checks if resource m is avail during period
                  MA(i, j) = 0
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If TAJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If

                      Next
                    Next
                  Next

                  'If M still 0 then m is available assign resource m to task j
938:                If MA(i, j) = 0 Then

                    TAMJ(i, j) = m 'Assign resource m to proj i task j
                    TFN(i, j) = TST(i, j) + TDM(i, j)
                    For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
```

```
                        TAJT(i, j, t) = m 'assign m to task k in period t
                      Next

                      'Forward Pass to Update TES and TEF
                      MAXEFPRC(i) = 0
                      For jj = j To JT
                        TES(i, jj) = TST(i, jj) - 1
                        TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                        For jr = 1 To JT
                          If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                            MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                          End If
                          If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then

                            TES(i, jj) = TEF(i, jr)
                            TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                          End If
                        Next
                      Next

                      GoTo Delayedtask3 'Redirect to next task

                    End If      '938

                  End If        '937 if FJ(i,j)not equal GM(i,j) go to next resource

                Next            '936 next resource

              Loop              '935 End lag loop

            End If              '925
Delayedtask3:
        Next                    '924 Iterate next task

      Next                      '922 Iterate next project


    'Compute Square Root of Sum of Squares
940:  For i = 1 To NP
      For j = 1 To JT
        TSE(i, j) = (TDL(i, j) - TDM(i, j)) ^ 2
      Next
    Next

    For i = 1 To NP
      SV(i) = 0
```

```
        Next

        For i = 1 To NP

            For j = 1 To JT
                SV(i) = SV(i) + (TSE(i, j))
                TSS(i) = Sqr(SV(i))
            Next
        Next


941: For i = 1 To NP
            MAXTEFPRC(i) = 0
             For jr = 1 To JT
                    If PRC(i, jr, JT) = 1 And TEF(i, jr) >= MAXTEFPRC(i) Then 'it is a
predecessor
                        MAXTEFPRC(i) = TEF(i, jr) 'set it to the max constant
                    End If
             Next
         Next

     'Complete Sequential Scheduling Method - Resource Constrained Solution
942:    For i = 1 To NP
            TST(i, JT) = MAXTEFPRC(i)      'Last node start equals SES (updated)
            TFN(i, JT) = MAXTEFPRC(i)      'Last node finish equals SEF (updated)

            For j = 2 To JT - 1
                If TAMJ(i, j) = 0 Then   'If not completely resourced, set schedule to null
                    TST(i, JT) = 0       'so that no value is attributed to that project
                    TFN(i, JT) = 0
                    TBF(i) = 0
                    TDE(i) = 0
                End If

            Next
            TCP(i) = TFN(i, JT)          'Otherwise complete the schedule
            TBB(i) = 0.1 * TCP(i)
            TBF(i) = TSS(i) + TBB(i)
            TTE(i) = TCP(i) + TBF(i)
            TDE(i) = TTE(i) - TL(i)
            If DE(i) < 0 Then
                TDE(i) = 0
            End If
        Next
```

```
    'Compute VE and NPVE for the projects that are scheduled
943:  For i = 1 To NP
      TVE(i) = 0
      TNPVE(i) = 0
      If TTE(i) > 0 Then

        'Compute TVE
        TVE(i) = VP(i) * (1 - ((3 * TDE(i)) / LP(i)) + ((2 * (TDE(i) ^ 2)) / (LP(i) ^ 2)))

        'Compute Expected NPV
        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
        REK(i) = RPK(i) * (1 - (2 * (TDE(i) / LP(i))))
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - TDE(i))) - (IR * ((LP(i) / 2) - TDE(i))) -
1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - TDE(i))))
        DFPFLE(i) = (1 + IR) ^ -TTE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
        TNPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

      End If
    Next '775

    'Compute Total Net Present Value Expected TNPVE
    TSUMNPVE = 0
    For ii = 1 To NP
      TSUMNPVE = TSUMNPVE + TNPVE(ii)
    Next

    'Compute Temporary Program Schedule Efficiency
    TPSE = TSUMNPVE / SUMNPVP


    ' If TPSE > PSE then make the switch, else keep the original assignments
980:  If TPSE > UPSE Then
      UPSE = TPSE
      USUMNPVE = TSUMNPVE
      SLAM12(MR) = UPSE

SUB12 = SUB12 + 1

      'make substitution of temporary assignments in solution
      For i = 1 To NP
    UCP(i) = TCP(i)
    UBF(i) = TBF(i)
    UTE(i) = TTE(i)
    UDE(i) = TDE(i)
    UNPVE(i) = TNPVE(i)

     For j = 1 To JT
       UDM(i, j) = TDM(i, j)
       UDL(i, j) = TDL(i, j)
       UAMJ(i, j) = TAMJ(i, j)
       UAHJ(i, j) = TAHJ(i, j)
       UES(i, j) = TES(i, j)
       UEF(i, j) = TEF(i, j)
       ULS(i, j) = TLS(i, j)
       ULF(i, j) = TLF(i, j)
       UST(i, j) = TST(i, j)
       UFN(i, j) = TFN(i, j)
       UTS(i, j) = TTS(i, j)
       UWS(i, j) = TWS(i, j)
       USC(i, j) = TSC(i, j)
       ULG(i, j) = TLG(i, j)

       For t = 1 To TP
         UAJT(i, j, t) = TAJT(i, j, t)
       Next
    Next
  Next

  GoTo STARTOVER12 'Start the search over

Else  ' If no substitution,Set Trial solution back for both x and y tasks


        TDM(ix, jx) = UDM(ix, jx)
        TDL(ix, jx) = UDL(ix, jx)
        TAHJ(ix, jx) = 1

        TDM(iy, jy) = UDM(iy, jy)
        TDL(iy, jy) = UDL(iy, jy)
        TAHJ(iy, jy) = 2

        For i = 1 To NP
          For j = 1 To JT

            TES(i, j) = UES(i, j)
            TEF(i, j) = UEF(i, j)
```

```
                                TLS(i, j) = ULS(i, j)
                                TLF(i, j) = ULF(i, j)
                                TST(i, j) = UST(i, j)
                                TFN(i, j) = UFN(i, j)
                                TCT(i, j) = UCT(i, j)
                                TTS(i, j) = UTS(i, j)

                                TWS(i, j) = UWS(i, j)
                                TSC(i, j) = USC(i, j)
                                TLG(i, j) = ULG(i, j)

                                TAMJ(i, j) = UAMJ(i, j)
                                TAHJ(i, j) = UAHJ(i, j)


                                For t = 1 To TP
                                    TAJT(i, j, t) = UAJT(i, j, t)
                                Next
                            Next
                        Next

                    End If      '980 End Trial Condition

                End If          '910  End Trial Solution

            Next                '909 Try Next task jy

        Next                    '908 Try Next project iy

        End If                  '907 End if hx

     Next                       '906 Try Next task jx

  Next                          '905 Try Next project ix


        'PRINT  SKILL-LEVEL ASSIGNMENT METHOD (SLAM) - Level 1 : 2
Substitution
        'Format Variables for printing, this preserves the original solution
990:   If MMR = 0 Then     'Print only if a single resource set is run

        PSEO = UPSE
        SUMNPVEO = USUMNPVE
        SUMNPVPO = SUMNPVP
```

```
    For i = 1 To NP
      CPO(i) = UCP(i)
      BFO(i) = UBF(i)
      TEO(i) = UTE(i)
      DEO(i) = UDE(i)
      NPVEO(i) = UNPVE(i)

      For j = 1 To JT
        DMO(i, j) = UDM(i, j)
        DLO(i, j) = UDL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = UES(i, j)
        EFO(i, j) = UEF(i, j)
        LSO(i, j) = ULS(i, j)
        LFO(i, j) = ULF(i, j)
        STO(i, j) = UST(i, j)
        FNO(i, j) = UFN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = UTS(i, j)
        FSO(i, j) = UFS(i, j)
        WSO(i, j) = UWS(i, j)
        SCO(i, j) = USC(i, j)
        LGO(i, j) = ULG(i, j)
        AMJO(i, j) = UAMJ(i, j)
        AHJO(i, j) = UAHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = UAJT(i, j, t)
        Next
      Next
    Next


991:  'Print Results of the Level Scheduling Method
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "900 - SKILL LEVEL
ASSIGNMENT METHOD (SLAM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "LEVEL 1:2 SUBSTITUTION
RESULTS"
    Worksheets("Results").Cells(Row + 2, Col).Value = "1:2 Substitution Trials:"
    Worksheets("Results").Cells(Row + 2, Col + 5).Value = TRIAL12
    Worksheets("Results").Cells(Row + 3, Col).Value = "1:2 Substitutions:"
    Worksheets("Results").Cells(Row + 3, Col + 5).Value = SUB12

    Row = Row + 4
```

```
     Call Print_Results

     End If     '990

999:   'End 900-999 Level12 Assignment Method

End Sub

Sub Level23_Assignment_Method()

1000:  'SKILL LEVEL ASSIGNMENT METHOD (SLAM)
     'This procedure iteratively completes trial solutions of schedules
     'where resources of same type but differing levels are interchanged.
     'This procedures specifically interchanges Level 2 and Level 3.
     'The procedure identifies pairings, exchanges resources including work
     'rates and then reassigns all other resources holding the pairing fixed.
     'Trial solution Total NPV is compared with the best SLAM solution and
     'updated if the results are improved.
     'This procedure is intitialized with the results of the solutions
     'from procedure(900) Level12 Assignement Method.

1001     'Declare Local Variables

     Dim ix As Integer          'Index of projects
     Dim jx As Integer          'Index of tasks
     Dim iy As Integer          'Index of projects
     Dim jy As Integer          'Index of tasks
     Dim nz As Integer           'Index of projects
     Dim hx As Integer           'Skill level of x
     Dim hy As Integer           'Skill level of y
     Dim my As Integer            'Resource id of x
     Dim mx As Integer            'Resource id of y
     Dim gy As Integer           'Resource type of x
     Dim gx As Integer           'Resource type of y

     Dim TPSE As Single         'Program Schedule Efficiency
     Dim TSUMNPVE As Single     'Total Sum of Program Expected Net Present Value
     Dim TCP(6) As Integer       'Critical Path
     Dim TBB(6) As Single         'Bias Buffer
     Dim TBF(6) As Single        'Total Buffer
     Dim TCC(6) As Integer        'Critical Chain Duration
     Dim TTE(6) As Integer       'Expected project duration, equivalent to CCD
     Dim TDE(6) As Integer        'Expected Delay between potential launch and expected
duration
     Dim TSS(6) As Integer       'Sum of Square Errors
```

```
    Dim TVE(6) As Variant       'Expected Value
    Dim TNPVE(6) As Single       'Expected Net Present Value
    Dim TDM(6, 10) As Integer  'Mean task duration estimates
    Dim TDL(6, 10) As Integer   'Low risk task duration estimates
    Dim TES(6, 10) As Integer   'Earliest Start Time task j
    Dim TEF(6, 10) As Integer   'Earliest Finish Time task j
    Dim TLS(6, 10) As Integer   'Late Start Time task j
    Dim TLF(6, 10) As Integer   'Late Finish Time task j
    Dim TTS(6, 10) As Integer   'Total slack
    Dim TWS(6, 10) As Integer   'Working slack
    Dim TSC(6, 10) As Integer   'Slack counter
    Dim TLG(6, 10) As Integer   'Lag
    Dim TCT(6, 10) As Integer   'Critical Path Task Indicator 1/0
    Dim TSE(6, 10) As Integer   'Square error
    Dim TST(6, 10) As Integer   'Start Time for task j
    Dim TFN(6, 10) As Integer   'Finish Time for task j
    Dim TAMJ(6, 10) As Integer 'Resource assignment matrix
    Dim TAHJ(6, 10) As Integer  'Skill level assignment matrix
    Dim TAJT(6, 10, 800) As Integer 'Assignment schedule matrix

    Dim TEMPX As Integer        'Temp Value of X for sorting
    Dim TEMPY As Integer        'Temp Value of Y for sorting
    Dim TEMPMX As Integer        'Resource x id substitution
    Dim TEMPMY As Integer        'Resource y id for substitution
    Dim TEMPHX As Integer        'Temporary Skill Level substitution
    Dim TEMPHY As Integer        'Temporary Skill Level substitution


1002:  'Initialize Solution Variables V - Equal to Level 1:2 SLAM Solution U
    VPSE = UPSE
    SLAM23(MR) = SLAM12(MR)

    For i = 1 To NP
      VCP(i) = UCP(i)
      VBF(i) = UBF(i)
      VBB(i) = UBB(i)
      VSS(i) = USS(i)
      VTE(i) = UTE(i)
      VDE(i) = UDE(i)
      VNPVE(i) = UNPVE(i)
      VVE(i) = UVE(i)

      For j = 1 To JT
        VDM(i, j) = UDM(i, j)
        VDL(i, j) = UDL(i, j)
        VES(i, j) = UES(i, j)
        VEF(i, j) = UEF(i, j)
        VLS(i, i) = ULS(i, j)
        ULF(i, j) = ULF(i, j)
        VST(i, j) = UST(i, j)
        VFN(i, j) = UFN(i, j)
        VTS(i, j) = UTS(i, j)
        VWS(i, j) = UWS(i, j)
        VSC(i, j) = USC(i, j)
        VLG(i, j) = ULG(i, j)
        VAMJ(i, j) = UAMJ(i, j)
        VAHJ(i, j) = UAHJ(i, j)

        For t = 1 To TP
          VAJT(i, j, t) = UAJT(i, j, t)
        Next
      Next
    Next


1003: 'Initialize Temporary Variables for Substitution Trials

    For i = 1 To NP
      TCP(i) = 0
      TBF(i) = 0
      TBB(i) = 0
      TSS(i) = 0
      TTE(i) = 0
      TDE(i) = 0
      TNPVE(i) = 0
      TVE(i) = 0

      For j = 1 To JT
        TDM(i, j) = VDM(i, j)
        TDL(i, j) = VDL(i, j)
        TES(i, j) = 0
        TEF(i, j) = 0
        TLS(i, i) = 0
        TLF(i, j) = 0
        TST(i, j) = VST(i, j)
        TFN(i, j) = VFN(i, j)
        TTS(i, j) = 0
        TWS(i, j) = 0
        TSC(i, j) = 0
        TLG(i, j) = 0

        TAMJ(i, j) = VAMJ(i, j)
```

```
                TAHJ(i, j) = VAHJ(i, j)

                For t = 1 To TP
                    TAJT(i, j, t) = VAJT(i, j, t)
                Next
            Next
        Next

    'Initialize Counters for substitution
    TRIAL23 = 0
    SUB23 = 0


STARTOVER23:

      'Starting at highest priority project and working to the lower priority projects
      'search through tasks sequentially for a level 1 resource assigned to that task
1005:  For n = 1 To NP

        If PP = 0 Then             'PP=0 for Sequentially ordered projects
            ix = n
            iy = n
        End If
        If PP = 1 Then             'PP=1 for PRI Productivity prioritized projects
            ix = PRI(n)
            iy = PRI(n)
        End If
        If PP = 2 Then             'PP=2 for NPVP Potential NPV prioritized projects
            ix = NPVPPRI(n)
            iy = NPVPPRI(n)
        End If
        If PP = 3 Then             'PP=3 for Random Project Order
            ix = RPRI(n)
            iy = RPRI(n)
        End If

        'Check resource at task jx
1006:        For jx = 2 To JT - 1
            mx = UAMJ(ix, jx)
            gx = GM(mx, MR)
            hx = HM(mx, MR)

            'Check if resource mx has Skill level hx
1007:           If hx = 2 Then     'if task ix,jx requires a level 1, then continue search

                'Check resources on projects iy
```

```
1008:       For iy = 1 To NP 'Search all projects for a Level 2

            'Check resources on tasks jy
1009:         For jy = 2 To JT - 1


            my = VAMJ(iy, jy) 'identify the resource for the task
            gy = GM(my, MR)
            hy = HM(my, MR)

            'Consider reassignment of x for y if:
            '1) Resource Type of x matches Resource Type of y
            '2) Skill Level of x is 1 and Skill level of y is 2
            '3) Task x and y occur in overlapping periods
            ' A fourth criteria may be considered later; TDM(x)< TDM(y)

            'Check if Resource Type and Skill Level match, and assigned in overlapping
time period
1010:           If gx = gy And hy = 3 And (TST(ix, jx) <= TFN(iy, jy) And TFN(ix, jx)
>= TST(iy, jy)) Then

TRIAL23 = TRIAL23 + 1   'Count number of trials

              'Trial solution alters durations and substitutes level of x and y tasks
              TDM(ix, jx) = VDM(ix, jx) / (1 - Q3)
              TDL(ix, jx) = VDL(ix, jx) / (1 - Q3)


              TDM(iy, jy) = VDM(iy, jy) * (1 - Q3)
              TDL(iy, jy) = VDL(iy, jy) * (1 - Q3)


1012:           'Forward Pass to Determine TES, TEF
              'Initialize TES(1) and TEF(1)
              For i = 1 To NP
                 TES(i, 1) = 0 + TLG(i, 1)
                 TEF(i, 1) = TDM(i, 1) + TES(i, 1)
              Next


              'The earliest start (TES)for any task is equal to the maximum of the
              'earliest finish(TEF)of the immediate predecessors
1013:            For i = 1 To NP
                 For j = 2 To JT 'start with task 2, search all rows in PRC for indicator
                    TES(i, j) = 0
                    For jr = 1 To JT
```

```
                      If PRC(i, jr, j) = 1 Then 'it is a predecessor
                         MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                      End If
                      If PRC(i, jr, j) = 1 And MAXEFPRC(i) >= TES(i, j) Then
                         TES(i, j) = TEF(i, jr) '+ LG(i, j)
                         TEF(i, j) = TES(i, j) + TDM(i, j)
                      End If
                   Next
                Next
             Next


1015:           'Backward Pass to Determine TLF, TLS
             'Initialize
             For i = 1 To NP
                TLF(i, JT) = TEF(i, JT)
                TLS(i, JT) = TLF(i, JT) - TDM(i, JT)
             Next

             'The latest finish (TLF) for any task is the smallest
             'of the latest start(TLS) times of immediate successors
             For i = 1 To NP
                For j = JT - 1 To 1 Step -1
                   TLF(i, j) = TLS(i, JT)
                   For jr = JT To 1 Step -1
                      If PRC(i, j, jr) = 1 Then
                         MINLSSUC(i) = TLS(i, jr)
                      End If
                      If PRC(i, j, jr) = 1 And MINLSSUC(i) <= TLF(i, j) Then
                         TLF(i, j) = TLS(i, jr) '- LG(i, jr) 'debug trial add LG at jr
                         TLS(i, j) = TLF(i, j) - TDM(i, j)
                      End If
                   Next
                Next
             Next

1016:        'Compute New Total Slack
          'Total Slack of activity j is the difference between 'the latest finish TLF(J)
          'and earliest finish EF(J),or the difference between TLS(J) and TES(J)
          For i = 1 To NP
             For j = 1 To JT
                TTS(i, j) = TLF(i, j) - TEF(i, j)
             Next
          Next
```

```
1020:           'Preassignment Process, Substitute and assign to specific tasks

            'Initialize TAMJ, clear matrix except resource substituton in x and y
            TEMPMX = VAMJ(ix, jx)
            TEMPMY = VAMJ(iy, jy)
            For i = 1 To NP
               For j = 1 To JT
                  TAMJ(i, j) = 0
               Next
            Next
            TAMJ(ix, jx) = TEMPMY
            TAMJ(iy, jy) = TEMPMX

            'Assign TAHJ, retain skill levels for subtitution x and y
            TEMPHX = VAHJ(ix, jx)
            TEMPHY = VAHJ(iy, jy)

            TAHJ(ix, jx) = TEMPHY
            TAHJ(iy, jy) = TEMPHX


            'Pre-assign TAJT for x and y
1021:           For i = 1 To NP
               For j = 1 To JT
                  For t = 1 To TP
                    TAJT(i, j, t) = 0 'reset TAJT on all tasks
                  Next
                  If i = ix And j = jx Then
                     For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                        TAJT(i, j, t) = TAMJ(ix, jx)
                     Next

                     ElseIf i = iy And j = jy Then
                        For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                           TAJT(i, j, t) = TAMJ(iy, jy)
                        Next
                  End If
               Next
            Next


            'Select Priority Order for Projects
1022:           For nz = 1 To NP
               If PP = 0 Then        'PP=0 for Sequentially ordered projects
                  i = nz
               End If
```

```
              If PP = 1 Then        'PP=1 for PRI Productivity prioritized projects
                i = PRI(nz)
              End If
              If PP = 2 Then        'PP=2 for NPVP Potential NPV prioritized projects
                i = NPVPPRI(nz)
              End If
              If PP = 3 Then        'PP=3 for Random Project Order
                i = RPRI(nz)
              End If

1024:           For j = 2 To JT - 1

1025:          If i = ix And j = jx Then          'Already assigned resource from y
                GoTo Delayedtask4
                ElseIf i = iy And j = jy Then  'Already assigned resource from x
                  GoTo Delayedtask4
                Else                       ' All other tasks to be assigned


              'Initialize ST
              TST(i, j) = TES(i, j) + 1

              'Try to directly assign a resource if available in required periods
1026:            For m = 1 To MR  'Assign resource sequentially one at a time

                'Check for both Resource Type and Skill Level match
1028:             If FJ(i, j) = GM(m, MR) And TAHJ(i, j) = HM(m, MR) Then

                  'Checks if resource m is avail during period
                  MA(i, j) = 0
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If TAJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If
                      Next
                    Next
                  Next

                  'If M equals 0 then m is available, assign to task j
                  If MA(i, j) = 0 Then
                    TAMJ(i, j) = m 'Assign resource m to proj i task j
                    TFN(i, j) = TST(i, j) + TDM(i, j) - 1 'Update Finish Date
                    For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                      TAJT(i, j, t) = m 'assign m to task k in period t
                      Next
                      GoTo Delayedtask4 'Redirect to next task
                    End If
                  End If      '1028
                Next          '1026 Try to assign all resources directly

                'If not assigned, use slack if available to delay task
1030:              Do While MA(i, j) = 1 And TSC(i, j) < TTS(i, j)

                  'Initialize
                  TSC(i, j) = TSC(i, j) + 1
                  TWS(i, j) = TTS(i, j) - TSC(i, j)
                  TST(i, j) = TES(i, j) + TSC(i, j)

                  'Assign resources one at a time
1031:                For m = 1 To MR

                    'Check for both Resource Type and Skill Level match
1032:                  If FJ(i, j) = GM(m, MR) And TAHJ(i, j) = HM(m, MR) Then

                      'Checks if resource is avail during period
                      MA(i, j) = 0
                        For t = ST(i, j) To ST(i, j) + TDM(i, j) - 1
                          For iii = 1 To NP
                            For jjj = 1 To JT
                              If TAJT(iii, jjj, t) = m Then
                                MA(i, j) = 1 'set to 1 if not avail
                              End If
                            Next
                          Next
                        Next

                        'If m is available in required interval, assign to j
1033:                      If MA(i, j) = 0 Then

                          TAMJ(i, j) = m 'Assign resource m to proj i task j
                          TFN(i, j) = TST(i, j) + TDM(i, j)

                          For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                            TAJT(i, j, t) = m 'assign m to task k in period t
                          Next

                          'Forward Pass to Update TES and TEF
                          MAXEFPRC(i) = 0
                          For jj = j To JT
                            TES(i, jj) = TST(i, jj) - 1
                      TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                      For jr = 1 To JT
                        If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                          MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                        End If
                        If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then

                          TES(i, jj) = TEF(i, jr)
                          TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                        End If
                      Next
                    Next
                    GoTo Delayedtask4 'Redirect to next task
                  End If    '1033
                End If      '1032
              Next          '1031 Try all resources at each level of slack
            Loop            '1030 End do while loop to utilize slack

            'If still not assigned increment lag until a resource is free
1035:          Do While TAMJ(i, j) = 0

            'Initialize
            TLG(i, j) = TLG(i, j) + 1
            TST(i, j) = TES(i, j) + TSC(i, j) + TLG(i, j)

              'Assign resource sequentially one at a time
1036:            For m = 1 To MR

                'Check for both Resource Type and Skill Level match
1037:              If FJ(i, j) = GM(m, MR) And TAHJ(i, j) = HM(m, MR) Then

                  'Checks if resource m is avail during period
                  MA(i, j) = 0
                  For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                    For ii = 1 To NP
                      For jj = 1 To JT
                        If TAJT(ii, jj, t) = m Then
                          MA(i, j) = 1 'set to 1 if not avail
                        End If

                      Next
                    Next
                  Next

                  'If M still 0 then m is available assign resource m to task j
1038:                If MA(i, j) = 0 Then
```

```
                      TAMJ(i, j) = m 'Assign resource m to proj i task j
                      TFN(i, j) = TST(i, j) + TDM(i, j)
                      For t = TST(i, j) To TST(i, j) + TDM(i, j) - 1
                        TAJT(i, j, t) = m 'assign m to task k in period t
                      Next

                      'Forward Pass to Update TES and ETF
                      MAXEFPRC(i) = 0
                      For jj = j To JT
                        TES(i, jj) = TST(i, jj) - 1
                        TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                        For jr = 1 To JT
                          If PRC(i, jr, jj) = 1 Then 'it is a predecessor
                            MAXEFPRC(i) = TEF(i, jr) 'set it to the max constant
                          End If
                          If PRC(i, jr, jj) = 1 And MAXEFPRC(i) >= TES(i, jj)
Then
                            TES(i, jj) = TEF(i, jr)
                            TEF(i, jj) = TES(i, jj) + TDM(i, jj)
                          End If
                        Next
                      Next
                      GoTo Delayedtask4 'Redirect to next task
                    End If  '1038
                  End If    '1037 if FJ(i,j)not equal GM(i,j) go to next resource
                Next        '1036 Try Next resource
              Loop          '1035 End lag loop
           End If           '1025
Delayedtask4:
         Next               '1024 Iterate next task
       Next                 '1022 Iterate next project


    'Compute Square Root of Sum of Squares
1040:  For i = 1 To NP
      For j = 1 To JT
        TSE(i, j) = (TDL(i, j) - TDM(i, j)) ^ 2
      Next
    Next

    For i = 1 To NP
      SV(i) = 0
    Next

    For i = 1 To NP
```

```
        For j = 1 To JT
          SV(i) = SV(i) + (TSE(i, j))
          TSS(i) = Sqr(SV(i))
        Next
      Next

     'Determine maximum TEF of last task predecessors
1041:  For i = 1 To NP
        MAXTEFPRC(i) = 0
         For jr = 1 To JT
            If PRC(i, jr, JT) = 1 And TEF(i, jr) >= MAXTEFPRC(i) Then 'it is a
predecessor
              MAXTEFPRC(i) = TEF(i, jr) 'set it to the max constant
            End If
         Next
       Next

    'Complete Sequential Scheduling Method - Resource Constrained Solution
1042:   For i = 1 To NP
        TST(i, JT) = MAXTEFPRC(i)     'Last node start equals SES (updated)
        TFN(i, JT) = MAXTEFPRC(i)     'Last node finish equals SEF (updated)

        For j = 2 To JT - 1
          If TAMJ(i, j) = 0 Then  'If not completely resourced, set schedule to null
            TST(i, JT) = 0      'so that no value is attributed to that project
            TFN(i, JT) = 0
            TBF(i) = 0
            TDE(i) = 0
          End If

        Next
        TCP(i) = TFN(i, JT)         'Otherwise complete the schedule
        TBB(i) = 0.1 * TCP(i)
        TBF(i) = TSS(i) + TBB(i)
        TTE(i) = TCP(i) + TBF(i)
        TDE(i) = TTE(i) - TL(i)
        If DE(i) < 0 Then
          TDE(i) = 0
        End If
      Next


      'Compute VE and NPVE for the projects that are scheduled
1043:    For i = 1 To NP
        TVE(i) = 0
        TNPVE(i) = 0
      If TTE(i) > 0 Then

        TVE(i) = VP(i) * (1 - ((3 * TDE(i)) / LP(i)) + ((2 * (TDE(i) ^ 2)) / (LP(i) ^ 2)))

        'Compute Expected NPV
        GEL(i) = (4 * VP(i)) / (LP(i) ^ 2)
        REK(i) = RPK(i) * (1 - (2 * (TDE(i) / LP(i))))
        GER(i) = REK(i) / (LP(i) / 2)
        DFPGLE(i) = (((1 + IR) ^ ((LP(i) / 2) - TDE(i))) - (IR * ((LP(i) / 2) - TDE(i))) -
1) / ((IR ^ 2) * ((1 + IR) ^ ((LP(i) / 2) - TDE(i))))
        DFPFLE(i) = (1 + IR) ^ -TTE(i)
        DFPARE(i) = (((1 + IR) ^ (LP(i) / 2)) - 1) / (IR * ((1 + IR) ^ (LP(i) / 2)))
        DFPGRE(i) = ((((1 + IR) ^ (LP(i) / 2)) - (IR * (LP(i) / 2)) - 1) / ((IR ^ 2) * ((1 +
IR) ^ (LP(i) / 2))))
        DFPFRE(i) = ((1 + IR) ^ (-(TL(i) + (LP(i) / 2))))
        TNPVE(i) = GEL(i) * DFPGLE(i) * DFPFLE(i) + ((REK(i) * DFPARE(i)) -
(GER(i) * DFPGRE(i))) * DFPFRE(i)

      End If
    Next '775

    'Compute Total Net Present Value Expected TNPVE
    TSUMNPVE = 0
    For ii = 1 To NP
      TSUMNPVE = TSUMNPVE + TNPVE(ii)
    Next

    'Compute Temporary Program Schedule Efficiency
    TPSE = TSUMNPVE / SUMNPVP


    ' If TPSE > PSE then make the switch, else keep the original assignments
1080:  If TPSE > VPSE Then
      VPSE = TPSE
      VSUMNPVE = TSUMNPVE
      SLAM23(MR) = VPSE

SUB23 = SUB23 + 1

      'make substitution of temporary assignments in solution
      For i = 1 To NP
        VCP(i) = TCP(i)
        VBF(i) = TBF(i)
        VTE(i) = TTE(i)
        VDE(i) = TDE(i)
        VNPVE(i) = TNPVE(i)
        For j = 1 To JT
          VDM(i, j) = TDM(i, j)
          VDL(i, j) = TDL(i, j)
          VAMJ(i, j) = TAMJ(i, j)
          VAHJ(i, j) = TAHJ(i, j)
          VES(i, j) = TES(i, j)
          VEF(i, j) = TEF(i, j)
          VLS(i, j) = TLS(i, j)
          VLF(i, j) = TLF(i, j)
          VST(i, j) = TST(i, j)
          VFN(i, j) = TFN(i, j)
          VTS(i, j) = TTS(i, j)
          VWS(i, j) = TWS(i, j)
          VSC(i, j) = TSC(i, j)
          VLG(i, j) = TLG(i, j)

          For t = 1 To TP
             VAJT(i, j, t) = TAJT(i, j, t)
          Next
        Next
     Next

     GoTo STARTOVER23 'Start the search over

  Else  ' If no substitution,Set Trial solution back for both x and y tasks

             TDM(ix, jx) = VDM(ix, jx)
             TDL(ix, jx) = VDL(ix, jx)
             TAHJ(ix, jx) = 2

             TDM(iy, jy) = VDM(iy, jy)
             TDL(iy, jy) = VDL(iy, jy)
             TAHJ(iy, jy) = 3

             For i = 1 To NP
                For j = 1 To JT

                   TES(i, j) = VES(i, j)
                   TEF(i, j) = VEF(i, j)
                   TLS(i, j) = VLS(i, j)
                   TLF(i, j) = VLF(i, j)
                   TST(i, j) = VST(i, j)
                   TFN(i, j) = VFN(i, j)
                   TCT(i, j) = VCT(i, j)
                   TTS(i, j) = VTS(i, j)
                TWS(i, j) = VWS(i, j)
                TSC(i, j) = VSC(i, j)
                TLG(i, j) = VLG(i, j)
                TAMJ(i, j) = VAMJ(i, j)
                TAHJ(i, j) = VAHJ(i, j)

                For t = 1 To TP
                  TAJT(i, j, t) = VAJT(i, j, t)
                Next
              Next
            Next
          End If '1080 End Trial condition, update
        End If   '1010 End Trial Solution
      Next       '1009 Try Next task jy
    Next         '1008 Try Next project iy
    End If       '1007 If hx = 2
  Next           '1006 Try Next task jx
Next             '1005 Try Next project ix


    'PRINT  SKILL-LEVEL ASSIGNMENT METHOD (SLAM) - Level 1 : 2
Substitution
    'Format Variables for printing, this preserves the original solution
1090:  If MMR = 0 Then    'Print only if a single resource set is run

    PSEO = VPSE
    SUMNPVEO = USUMNPVE
    SUMNPVPO = SUMNPVP

    For i = 1 To NP
      CPO(i) = VCP(i)
      BFO(i) = VBF(i)
      TEO(i) = VTE(i)
      DEO(i) = VDE(i)
      NPVEO(i) = VNPVE(i)

      For j = 1 To JT
        DMO(i, j) = VDM(i, j)
        DLO(i, j) = VDL(i, j)
        FJO(i, j) = FJ(i, j)
        ESO(i, j) = VES(i, j)
        EFO(i, j) = VEF(i, j)
        LSO(i, j) = VLS(i, j)
        LFO(i, j) = VLF(i, j)
        STO(i, j) = VST(i, j)
        FNO(i, j) = VFN(i, j)
        CTO(i, j) = CT(i, j)
        TSO(i, j) = VTS(i, j)
        FSO(i, j) = VFS(i, j)
        WSO(i, j) = VWS(i, j)
        SCO(i, j) = VSC(i, j)
        LGO(i, j) = VLG(i, j)
        AMJO(i, j) = VAMJ(i, j)
        AHJO(i, j) = VAHJ(i, j)

        For t = 1 To TP
          AJTO(i, j, t) = VAJT(i, j, t)
        Next
      Next
    Next


1091:   'Print Results of the Level Scheduling Method
    Row = Resultsrow
    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "1000 - SKILL LEVEL
ASSIGNMENT METHOD (SLAM)"
    Worksheets("Results").Cells(Row + 1, Col).Value = "LEVEL 2:3 SUBSTITUTION
RESULTS "
    Worksheets("Results").Cells(Row + 2, Col).Value = "2:3 Substitution Trials:"
    Worksheets("Results").Cells(Row + 2, Col + 5).Value = TRIAL23
    Worksheets("Results").Cells(Row + 3, Col).Value = "2:3 Substitutions:"
    Worksheets("Results").Cells(Row + 3, Col + 5).Value = SUB23

    Row = Row + 4

    Call Print_Results

    End If     '1090

1099    'End 1000-1099  Level 23 Assignment Method
    End Sub



    Sub Print_Results()

    Dim TT As Integer   'Schedule will only be printed to last task TT

    Col = Excelcol
    Worksheets("Results").Cells(Row, Col).Value = "Project Prioritization PP:"
    Worksheets("Results").Cells(Row, 6).Value = PP
    Worksheets("Results").Cells(Row + 1, Col).Value = "Resource Prioritization PR:"
    Worksheets("Results").Cells(Row + 1, 6).Value = PR
    Worksheets("Results").Cells(Row + 2, Col).Value = "Resource Quantity MR:"
    Worksheets("Results").Cells(Row + 2, 6).Value = MR

    'Print AMJ the matrix of assignments
    Row = Row + 4
    Col = 1
    For i = 1 To NP
      For j = 1 To JT
        If i = 1 And j = 2 Then
          Worksheets("Results").Cells(Row, Col + 4 + j).Value = "Assignment Matrix
AMJ - Task"
        End If
        If i = 2 And j = 1 Then
           Worksheets("Results").Cells(Row + 1, Col).Value = "Project"
        End If
        If i = 1 Then
          Worksheets("Results").Cells(Row + i, Col + j).Value = j
         End If
         Worksheets("Results").Cells(Row + 1 + i, Col).Value = i
         Worksheets("Results").Cells(Row + 1 + i, Col + j).Value = AMJO(i, j)
      Next
    Next

    'Print Updated Results

    Row = Row + 10
    Col = 1
    Worksheets("Results").Cells(Row, Col).Value = "Project"
    Worksheets("Results").Cells(Row + 1, Col).Value = "Task"
    Worksheets("Results").Cells(Row + 2, Col).Value = "DM"
    Worksheets("Results").Cells(Row + 3, Col).Value = "DL"
    Worksheets("Results").Cells(Row + 4, Col).Value = "FJ"

    Worksheets("Results").Cells(Row + 6, Col).Value = "ES"
    Worksheets("Results").Cells(Row + 7, Col).Value = "EF"
    Worksheets("Results").Cells(Row + 8, Col).Value = "LS"
    Worksheets("Results").Cells(Row + 9, Col).Value = "LF"
    Worksheets("Results").Cells(Row + 10, Col).Value = "ST"
    Worksheets("Results").Cells(Row + 11, Col).Value = "FN"

    Worksheets("Results").Cells(Row + 13, Col).Value = "CT"
    Worksheets("Results").Cells(Row + 14, Col).Value = "TS"
    Worksheets("Results").Cells(Row + 15, Col).Value = "FS"
    Worksheets("Results").Cells(Row + 16, Col).Value = "WS"
Worksheets("Results").Cells(Row + 17, Col).Value = "SC"
Worksheets("Results").Cells(Row + 18, Col).Value = "LG"

Worksheets("Results").Cells(Row + 20, Col).Value = "AMJ"
Worksheets("Results").Cells(Row + 21, Col).Value = "Type"
Worksheets("Results").Cells(Row + 22, Col).Value = "Level"


Worksheets("Results").Cells(Row + 24, Col).Value = "CP"
Worksheets("Results").Cells(Row + 25, Col).Value = "BF"
Worksheets("Results").Cells(Row + 26, Col).Value = "TE"
Worksheets("Results").Cells(Row + 27, Col).Value = "DE"
Worksheets("Results").Cells(Row + 28, Col).Value = "NPVE"

Worksheets("Results").Cells(Row + 29, Col).Value = "TOTAL"
Worksheets("Results").Cells(Row + 30, Col).Value = "NPVE"
Worksheets("Results").Cells(Row + 31, Col).Value = "NPVP"
Worksheets("Results").Cells(Row + 32, Col).Value = "PSE"


For i = 1 To NP
  For j = 1 To JT
    Worksheets("Results").Cells(Row, Col + 1).Value = i
    Worksheets("Results").Cells(Row + 1, Col + 1).Value = j
    Worksheets("Results").Cells(Row + 2, Col + 1).Value = DMO(i, j)
    Worksheets("Results").Cells(Row + 3, Col + 1).Value = DLO(i, j)
    Worksheets("Results").Cells(Row + 4, Col + 1).Value = FJO(i, j)

    Worksheets("Results").Cells(Row + 6, Col + 1).Value = ESO(i, j)
    Worksheets("Results").Cells(Row + 7, Col + 1).Value = EFO(i, j)
    Worksheets("Results").Cells(Row + 8, Col + 1).Value = LSO(i, j)
    Worksheets("Results").Cells(Row + 9, Col + 1).Value = LFO(i, j)
    Worksheets("Results").Cells(Row + 10, Col + 1).Value = STO(i, j)
    Worksheets("Results").Cells(Row + 11, Col + 1).Value = FNO(i, j)

    Worksheets("Results").Cells(Row + 13, Col + 1).Value = CTO(i, j)
    Worksheets("Results").Cells(Row + 14, Col + 1).Value = TSO(i, j)
    Worksheets("Results").Cells(Row + 15, Col + 1).Value = FSO(i, j)

    Worksheets("Results").Cells(Row + 16, Col + 1).Value = WSO(i, j)
    Worksheets("Results").Cells(Row + 17, Col + 1).Value = SCO(i, j)
    Worksheets("Results").Cells(Row + 18, Col + 1).Value = LGO(i, j)

    Worksheets("Results").Cells(Row + 20, Col + 1).Value = AMJO(i, j)
    Worksheets("Results").Cells(Row + 21, Col + 1).Value = FJO(i, j)
```

```
        If AMJO(i, j) > 0 Then
          m = AMJO(i, j)
          Worksheets("Results").Cells(Row + 22, Col + 1).Value = HM(m, MR)
        Else
          Worksheets("Results").Cells(Row + 22, Col + 1).Value = 0
        End If

        Col = Col + 1
      Next
    Next

    Col = Excelcol
    For i = 1 To NP
      Worksheets("Results").Cells(Row + 24, Col + 10).Value = CPO(i)
      Worksheets("Results").Cells(Row + 25, Col + 10).Value = BFO(i)
      Worksheets("Results").Cells(Row + 26, Col + 10).Value = TEO(i)
      Worksheets("Results").Cells(Row + 27, Col + 10).Value = DEO(i)
      Worksheets("Results").Cells(Row + 28, Col + 10).Value = NPVEO(i)

      Col = Col + 10
    Next
    Col = Excelcol
    Worksheets("Results").Cells(Row + 30, Col + 1).Value = SUMNPVEO
    Worksheets("Results").Cells(Row + 31, Col + 1).Value = SUMNPVPO
    Worksheets("Results").Cells(Row + 32, Col + 1).Value = PSEO


    "" Print Resource Information Each Program
    Col = 1
    Worksheets("Results").Cells(Row + 35, Col).Value = "RESOURCE SET"
    Worksheets("Results").Cells(Row + 36, Col).Value = "SRI - Sequential Resource
Index"
    Worksheets("Results").Cells(Row + 37, Col).Value = "GM - Resource Type"
    Worksheets("Results").Cells(Row + 38, Col).Value = "HM - Skill Level"
    Worksheets("Results").Cells(Row + 39, Col).Value = "LRI - Level Resource Index"

    'Worksheets("Results").Cells(Row + 40, Col).Value = "RRI - Random Resource
Index"

    Col = 10
    For m = 1 To MR
      Worksheets("Results").Cells(Row + 36, Col + m).Value = m
      Worksheets("Results").Cells(Row + 37, Col + m).Value = GM(m, MR)
      Worksheets("Results").Cells(Row + 38, Col + m).Value = HM(m, MR)
      Worksheets("Results").Cells(Row + 39, Col + m).Value = LRI(m, MR)
```

```
   'Worksheets("Results").Cells(Row + 40, Col + m).Value = RRI(m)

Next
Col = 1
Worksheets("Results").Cells(Row + 41, Col).Value = "GANTT CHART"
Worksheets("Results").Cells(Row + 42, Col).Value = "Project"
Worksheets("Results").Cells(Row + 43, Col).Value = "Task"
Worksheets("Results").Cells(Row + 44, Col).Value = "Day"

Col = 1
For i = 1 To NP
   For j = 1 To JT
      Worksheets("Results").Cells(Row + 42, Col + 1).Value = i
      Worksheets("Results").Cells(Row + 43, Col + 1).Value = j
      Col = Col + 1
   Next
Next

'Determine the maximum length project in the program TT
TT = 0
For i = 1 To NP
   If CPO(i) > TT Then
      TT = CPO(i)
   End If
Next



Col = 1
For i = 1 To NP
  For j = 1 To JT
      For t = 1 To TT
         Worksheets("Results").Cells(Row + 44 + t, 1).Value = t
         Worksheets("Results").Cells(Row + 44 + t, Col + j).Value = AJTO(i, j, t)
      Next
    Next
    Col = Col + 10
Next

"" RESET Variables for printing, this preserves the original solution

PSEO = 0
SUMNPVEO = 0
SUMNPVPO = 0
```

```
    For i = 1 To NP
      CPO(i) = 0
      BFO(i) = 0
      TEO(i) = 0
      DEO(i) = 0
      NPVEO(i) = 0

      For j = 1 To JT
        DMO(i, j) = 0
        DLO(i, j) = 0
        FJO(i, j) = 0
        ESO(i, j) = 0
        EFO(i, j) = 0
        LSO(i, j) = 0
        LFO(i, j) = 0
        STO(i, j) = 0
        FNO(i, j) = 0
        CTO(i, j) = 0
        TSO(i, j) = 0
        FSO(i, j) = 0
        WSO(i, j) = 0
        SCO(i, j) = 0
        LGO(i, j) = 0
        AMJO(i, j) = 0
        AHJO(i, j) = 0

        For t = 1 To TP
          AJTO(i, j, t) = 0
        Next
      Next
    Next

    Resultsrow = Resultsrow + 64 + TT

    '''' End Print Results Procedure

End Sub

Sub Print_Chart()

  Row = 1
  Col = 1

  Worksheets("Charts").Cells(Row, Col).Value = "MR"
  Worksheets("Charts").Cells(Row, Col + 1).Value = "CF"
  Worksheets("Charts").Cells(Row, Col + 2).Value = "SSM TCS"
```

```
 Worksheets("Charts").Cells(Row, Col + 3).Value = "SSM RCS"

 Worksheets("Charts").Cells(Row, Col + 4).Value = "CSM TCS"
 Worksheets("Charts").Cells(Row, Col + 5).Value = "CSM RCS"

 Worksheets("Charts").Cells(Row, Col + 6).Value = "ESM TCS"
 Worksheets("Charts").Cells(Row, Col + 7).Value = "ESM RCS"

 Worksheets("Charts").Cells(Row, Col + 8).Value = "BEST RCS"

 Worksheets("Charts").Cells(Row, Col + 9).Value = "LSM LCS"
 Worksheets("Charts").Cells(Row, Col + 10).Value = "SLAM 1:2"
 Worksheets("Charts").Cells(Row, Col + 11).Value = "SLAM 123"

 For m = KT To MRMAX
   Worksheets("Charts").Cells(Row + 1, Col).Value = m
   Worksheets("Charts").Cells(Row + 1, Col + 1).Value = CF(m)

   Worksheets("Charts").Cells(Row + 1, Col + 2).Value = SSMTCS(m)
   Worksheets("Charts").Cells(Row + 1, Col + 3).Value = SSMRCS(m)

   Worksheets("Charts").Cells(Row + 1, Col + 4).Value = CSMTCS(m)
   Worksheets("Charts").Cells(Row + 1, Col + 5).Value = CSMRCS(m)

   Worksheets("Charts").Cells(Row + 1, Col + 6).Value = ESMTCS(m)
   Worksheets("Charts").Cells(Row + 1, Col + 7).Value = ESMRCS(m)
   Worksheets("Charts").Cells(Row + 1, Col + 8).Value = BRCS(m)

   Worksheets("Charts").Cells(Row + 1, Col + 9).Value = LCS(m)
   Worksheets("Charts").Cells(Row + 1, Col + 10).Value = SLAM12(m)
   Worksheets("Charts").Cells(Row + 1, Col + 11).Value = SLAM23(m)


   Row = Row + 1
 Next

End Sub
```

REFERENCES

Afonso, P., Nunes, M., Paisana, A, Braga, A., (2008) "The Influence of time-to-market and target costing in the new product development process." International Journal of Production Economics. July 2008.

American Productivity & Quality Center (APQC). (2003). Improving New Product Development Performance and Practices, Houston, TX.

Adams-Bigelow, M. (2005). "2003 Comparative Performance Assessment Study." PDMA Handbook, 2nd ed.

Armstrong, G., Kotler, P. (2003). "Principles of Marketing." Prentice Hall, 10th ed.

Askin, R., Huang, Y. (2001). "Forming effective worker teams for cellular manufacturing." International Journal of Production Research. Vol. 39. No. 11.

Belliveau, P., Griffin, A., Somermeyer, S., (2002), "PDMA ToolBook for New Product Development, Wiley & Sons, 2002.

Belliveau, P., Griffin, A., Somermeyer, S., (2004), "PDMA ToolBook 2 for New Product Development, Wiley & Sons, 2004.

Bettis, R., Hitt, M. (1995). "The New Competitive Landscape." Strategic Management Journal.

Bock. D., Patterson, J. (1990). "A Comparison of Due Date Setting, Resource Assignment, and Job Preemption Heuristics for the Multiproject Scheduling Problem," Decision Sciences, Vol. 21, 387-402, 1990.

Bralla, J. G. (1996). "Design for Excellence." McGraw-Hill, Inc. pp 18-37.

Butler, J. Morrice, D., Mullarkey, P., (2001) "A Multiple Attribute Utility Theory Approach to Ranking and Selection." Management Science. June 2001.

Calantone, R., Di Benedetto, C., Schmidt, J. (1999),"Using the Analytic Hierarchy Process in New Product Screening." Vol. 16, No. 1, January 1999.

Chen, J., Askin, R.G, (2009). "Project Selection, scheduling and resource allocation with time dependent returns." European Journal of Operational Research. 193 – 23-34, 2009.

Chin, K., Xu, D., Yang, J, Lam, J., (2008) "Group-based ER-AHP system for product screening." Expert Systems with Applications. 35 1909-1929.

Cooper, R.G. (2005). “Product Leadership Pathways to Profitable Innovation.” Basic Books.

Cooper, R. G. (2005). “New Products – What Separates the Winners from the Losers and What Drives Success.” PDMA Handbook, 2nd ed.

Cooper, R.G. (2001). “Winning at New Products,” Basic Books.

Cooper, R. G., Edgett, S. J. and Kleinschmidt, E. J. (2001). “Portfolio Management for New Products, “ 2nd Edition pp. 3-14.

Crawford, M., Di Benedetto, A. (2003), “New Products Management.” 7th Ed. McGraw Hill.

Cusumano, M., Nobeoka, K., (1999), “Thinking Beyond Lean.” Free Press, 1998.

Davis, E., Patterson, J. (1975), "A Comparison of Heuristic and Optimum Solutions in Resource-Constrained project Scheduling." Management Science. Vol. 21, No.8, Spril 1975.

Davenport, T., Prusak, L. and Wilson, H. (2003). “Who’s Bringing You Hot Ideas?” Harvard Business Review. February.

Demeulemeester, E., Herroelen, W., (2002), Project Scheduling: A Research Handbook,” Kuwer Academic Publisheers, 2002.

Deming, W. Edwards (1986), “Out of the Crisis.” Massachusetts Institute of Technology, Center for Advanced Engineering Study. Cambridge, Mass.

DeReyck, B., Herroelen, W. (1998), “A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations,” European Journal of Operational Research, Vol. 111, 152-174, 1998.

Di Benedetto, A., (1999). “Identifying the Key Success Factors in New Product Launch.” Journal of Product Innovation Management.” Vol. 16, No. 5, November 1999.

Dickinson, N, Thornton, A, Graves, S., (2001), “Technology portfolio management: Optimizing interdependent projects over multiple time periods,” IEEE Engineering Management 48.

Dumond, J, Mabert, V. (1988). “Evaluating Project Scheduling and Due Date Assignment Procedures: An Experimental Analysis,” Management Science, Vol. 34, No. 1, January 1988.

Eppinger, S., Whitney, D., Smith, R., Gebala, D., (1994), "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design. Vol. 6, No.

Fendly, L., (1968) " Toward the Development of a Complete Multiproject Scheduling System," Journal of Industrial Engineering, Vol. 19, No. 10, October 1968.

Fox, G., Baker, N., Bryant, J. (1984), "Economic Models for R&D Project Selection in the Presence of Project Interactions," Management Science 30.

Fox, G., Baker, N. (1985), "Project Selection Decision Making Linked to a Dynamic Environment," Management Science 31.

Gatignon, H., Xuereb, J., (1997), "Strategic Orientation of the Firm and New Product Performance." Journal of Marketing Research. February.

Geekie, A., Steyn, H. (2008) "Buffer Sizing For the Critical Chain Project Management Method." South African Journal of Industrial Engineering, May 2008.

Gertler, M. (1995), "Being there: proximity, organization, and culture in the development and adoption of advanced manufacturing technologies," Economic Geography 71.

Griffin, A. (2002), "Product development cycle time for business-to-business products." Industrial Marketing Management. 31 (4) pp. 291-304.

Griffin, A. (1997), "The Effect of Project and Process Characteristics on Product Development Cycle Time." Journal of Marketing Research. February.

Griffin, A. (1997), "PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices." Journal of Product Innovation Management, November.

Griffin, A., and Hauser, J. (1996), "Integrating R & D and marketing: A review and analysis of the literature," Journal of Product Innovation Management.

Goldratt, E. (1997), "Critical Chain." The North River Press. Great Barrington, MA.

Goncalves, J., Mendes, J., Resende, M. (2008). "A genetic algorithm for the resource constrained multi-project scheduling problem," European Journal of Operational Research, Vol. 189, 1171-1190, 2008.

Haley, G., Goldberg, S., (1995) "New Present Value Techniques and Their Effects on New Product Research." Industrial Marketing Management, 24, 177-190.

Herroelen, W., Leus, R., (2001) "On the merits and pitfalls of critical chain scheduling." Journal of Operations Management, 19, 559-577.

Herroelen, W., and DeReyck, B. (1998), "A Branch and Bound Procedure for the resource-constrained project scheduling problem with generalized precedence relations," European Journal of Operational Research," 1998 (152-174).

Hillier, F., Lieberman, G. (2001). "Introduction to Operations Research." McGraw-Hill.

Hitt, M., Keats, B., DeMarie, S. (1998). "Navigating in the new competitive landscape: Building strategic flexibility and competitive advantage in the 21st century." Academy of Management Executive, 1988. Vol. 12. No. 4.

Huchzermeir, A., Loch, C. (2001), "Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D." Management Science. Vol. 47, No. 1.

IBM, (2006), ibm.com, "Revenue lost by being late to market."

Juran, J., and Godfrey, A. (1999), "Juran's Quality Handbook." Fifth Edition. McGraw Hill.

Kahn, K., (2005), "The PDMA Handbook of New Product Development," Wiley & Sons, 2005.

Kahney, L. (2004), "Inside Look at Birth of the IPod," Wired News. July 21, 2004.

Kahney, L. (2005), "The Cult of IPod," No Starch Press.

Kennedy, M. (2003), "Product Development for the Lean Enterprise." The Oaklea Press.

Kerzner, H. (2003), "Project Management: A Systems Approach to Planning, Scheduling and Controlling." John Wiley.

Klein, R., (2000), "Scheduling of Resource Constrained Projects," Kluwer Academic Publishers, 2000.

Koen, P. (2005), "The Fuzzy Front End for Incremental, Platform and Breakthrough Products." PDMA Handbook, 2nd ed.

Kolisch, R., and Hartmann, S., (2006) "Experimental investigation of heuristics for resource-constrained project scheduling: An Update," European Journal of Operational Research, 2006 (23-37).

Krishnan, V., Ulrich, K. (2001), "Product Development Decisions: A Review of the Literature," Management Science 47.

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

Kurtulus, I., Davis, E., (1982). "Multi-Project Scheduling: Categorization of Heuristic Rules Performance," Management Science, Vol. 28. No 2, February 1982.

Kurtulus, I., Narulla, S. (1985). "Multi-Project Scheduling: Analysis of Project Performance," IIE Transactions, Vol. 17, No. 1, 58-66, 1985.

Lawrence, S., Morton, T. (1993). "Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics," European Journal of Operational Research, Vol. 64, 168-187, 1993.

Leach, L. (2003-1) ""Simplified Critical Chain Project Management." Downloaded from www.advanced-products.com, January 2003.

Leach, L. (2003-2 "Schedule and Cost Buffer Sizing: How to Account for the Bias Between Project Performance and Your Model." Project Management Journal, June.

Lechler, T., Ronen, B., Stohr, E., (2005) "Critical Chain: A New Project Management Paradigm or Old Wine in New Bottles?" Engineering Management Journal. Vol. 17. No. 4 December 2005.

Lewis, M., Alidaee, B, Kochenberger, G., "Modeling and Solving the Task Allocation Problem As an Unconstrained Quadratic Binary Program." olemiss.edu.

Liu, J. (1995) "Detailed model shows FPGAs' true costs." Electronic Design News, May 11, 1995. pp. 153-156.

Loch, C., Kavadias, S., (2002) "Dynamic Portfolio Selection of NPD Programs Using Marginal Returns," Management Science. Vol. 48. No 10, October pp. 1227-1241.

Loch, C., Pich, M, Tewiesch, C, Urbschat, M.(2001), "Selecting R&D projects at BMW: A case study of adopting mathematical programming models," IEEE Engineering Management 48.

Lova, A., Maroto, C., Tormos, P. (2000). "A Multicriteria heuristic method to improve resource allocation in multiproject scheduling," European Journal of Operational Research, Vol 127, 408-424, 2000.

MacCormack, A., Iansite, M. (1997), "Developing Products on Internet Time." Harvard Business Review. Sept-Oct.

McGrath, M., (2001), "Product Strategy for High Technology Companies, McGraw-Hill, 2001.

Metcalf, M. (2006), "Closing the Chasm: Creating Successful Product Development Organizations." Manufacturing Net. May 4, 2006.

More, R.A. (1986), "Developer/adopter relationships in new industrial product situations." Journal of Business Research.

Morgan, J., Liker, J., (2006), "The Toyota Product Development System," Productivity Press. 2006.

Norman, B., Tharmmaphornphilas, W., Needy, K., Bidanda, B., Warner, R. (2002)."Worker assignment in cellular manufacturing considering technical and human skills." International Journal of Production Research. Vol. 40. No. 6.

Project Management Body of Knowledge. (2008), Fourth Ed., PMI Publications, 2008.

Pyzdek, T. (2003). "The Six Sigma Handbook." McGraw Hill.

Pu, Z. (2009), "NPD Project Profile Description and Analysis." NJIT Research Forum, July 2009.

R&D Magazine (2008), "2009 Global R&D Forecast," December, 2008.

Reagan, R. (2004). "It's Happening at the Hub." FTTH Conference Proceedings.

Reagan, R. (2005). "It's Still Happening at the Hub." FTTH Conference Proceedings.

Reynolds, R. (2006). "Five Laws of Product Development for Education in the 21st Century." Xplananazine.com. Februry 13, 2006.

Rehemtulla, M., Huges, G. (2006), "Meeting the Product Lifecycle Challenge." IBM Pub.

Ross, S. (1995), "Uses, Abuses, and Alternatives to the Net-Present-Value Rule," Financial Management, Vol. 24, No.3, Autumn 1995.

Sassenburg, H., Berghout, E. (2005) "The NPVI Method to Support Market Entry Strategies for Sofware Products."

Schwindt, C., (2005) "Resource Allocation in Project Management," Springer Publications, 2005.

Schmidt, R. (1993), "Model for R & D project selection with combined benefit, outcome and resource interactions," IEEE Engineering Management 40.

Sherman, E.,(2002). "Inside the Apple iPod Design Triumph." Electronics Design Chain. Summer 2002.

Smith, P., Reinertsen, D. (1998). "Developing Products in Half the Time." Wiley & Sons, 1998.

Smith, P., Merritt, G., (2002), "Proactive Risk Management." Productivity, 2002.

Tukel, O. Rom, W. Duni Eksioglu, S. (2006) "An Investigation of buffer sizing techniques in critical chain scheduling." European Journal of Operational Research, 401-416.

Ulrich, K., Eppinger, S. (2004), "Product Design and Development." McGraw Hill.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., (2008) "Proactive heuristic procedures for robust project scheduling: An experimental analysis," European Journal of Operational Research, Vol 189, 723-733, 2008.

Womack, J., Jones, D and Roos, D. (1990). "The Machine That Changed the World." Rawson Associates.

Xu, N., McKee, S., Nozick, L., Ufomata, R. (2008), "Augmenting priority rule heuristics with justification and rollout to solve the resource constrained project scheduling problem," Computers and Operations Research, Vol. 35, 3284-3297, 2008.