To the Graduate Council:

I am submitting herewith a thesis written by Vincent Gilles Delgado entitled "A Computer Modeling Approach Using Critical Resource Diagramming Network Analysis in Project Scheduling." I have examined the final paper copy of this thesis for form and content and recommend that it may be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Adedeji B. Badiru
Major Professor

We have read this thesis
and recommend its acceptance:

Mandyam Srinivasan

Fong-Yuen Ding

Accepted for the Council:

Vice Chancellor and
Dean of Graduate Studies

A COMPUTER MODELING APPROACH USING CRITICAL RESOURCE DIAGRAMMING NETWORK ANALYSIS IN PROJECT SCHEDULING

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Vincent Gilles Delgado

December 2004

Thesis
2004
.D44

Copyright © 2004 by Vincent Gilles Delgado
All rights reserved.

DEDICATION

This thesis is dedicated to

My parents, Felix Delgado and Marie-Jose Fernandez, and my sisters Marie-Yolande Delgado, Sandrine Delgado et Gisele Delgado for being there for me each day of my life.

My family and friends here in the United States and back home in Senegal

My aunt Marie-Madeleine Spencer and her children Roselyne and Nilton for making me feel at home, for always believing in me and encouraging me to reach higher in order to achieve my goals.

My grandmother Maria-Yolanda, my godmother Maria Daluz Andrade and my grandfather Edwino Monteiro for looking after me each day of my life

" A vous tous je vous aimerez toujours"

Acknowledgments

I wish to thank all those who helped me to complete my Master of Science degree in Industrial Engineering. I would especially like to thank Dr. Adedeji Badiru for his guidance and support through my education experience. His valued advice, guidance, and encouragement were influential in completing this thesis.

I would also like to thank Dr. Mandyam Srinivasan and Dr. Fong-Yuen Ding for serving on my committee and guiding me with great advice.

A special thank for John-Joseph Roberts Niemann for guiding me through the programming portion.

I would like to thank my family and friends whose suggestions, encouragement, love, and support kept me going through school and the hard times in life.

Lastly I wish to thank Jeannette Myers, for helping me and advising me during the elaboration of this work.

Abstract

The problem of resource constrained project scheduling (RCPSP) continues to be an important topic in project management. Different scheduling processes have been introduced to solve cases of RCPSP. Most of the developed methods are based on a network analysis approach. The two main technique of project network analysis used for planning, scheduling, and control are PERT and CPM. These approaches assume unlimited resource availability in project network analysis. In realistic projects, both the time and resource requirements of activities should be considered in developing network schedules. Another particularity of the methods created, so far, is the focus on activities during the scheduling process. Therefore, from a resource point of view, the current procedures do not allow the project manager to incorporate information concerning each resource unit under supervision in the scheduling process of a project. There is a need for simple tools for resource planning, scheduling, tracking, and control.

Critical resource diagramming (CRD) is a relatively new resource management tool. CRD is a simple extension to the CPM technique developed for resource management purposes. Unlike activity networks, CRD uses nodes to represent each resource unit. Also, contrasting with activities, a resource unit may appear more than once in a CRD network, specifying all different tasks to which a particular unit is assigned. Similar to CPM, the same backward and forward computations may be performed to CRD.

The CRD method can also be used in solving RCPSP problems. This present study explores that advantage of CRD. The purposes are to develop a methodology, based on CRD, which allows its use in specific case of RCPSP, to implement the developed technique using a computer model, to conduct test to validate the CRD computer model, and then to investigate the advantages and disadvantages of the introduced method. The CRD computer model will be implemented using the Visual Basic 6.0 language.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	INTRODUCTION	1
	1.1 Historical Perspective on Project Scheduling	1
	1.2 Resource Scheduling	2
	1.3 The Statement of the Research	4
	1.3.1 Critical Path Method (CPM)	5
	1.3.2 Project Evaluation and Review Technique (PERT)	5
	1.3.3 Critical Resource Diagramming (CRD)	6
	1.3.4 Objectives of the Research	7
	1.4 Thesis Organization	7
2.	LITERATURE REVIEW	9
	2.1 The Resource-Constrained Project Scheduling Problem (RCPSP)	10
	2.1.1 The Problem	10
	2.1.2 Characteristics of the RCPSP	14
	a) Activities	14
	b) Resources	15
	c) Constraints and Objectives	15
	2.1.3 Instances of the RCPSP	16
	2.2 Different solutions to the RCPSP	17
	2.2.1 Exact Solution Methods	18
	a) CPM – PERT	18
	b) Mathematical Programming	19

c) Branch-and-Bound Method..... 20

d) Critical Chain..... 21

2.2.2 Heuristic Solution Methods 22

a) Constructive Heuristics..... 23

b) Improvement Heuristics..... 23

3. FUNDAMENTALS OF NETWORK ANALYSIS..... 26

3.1 Project Network Analysis..... 27

3.2 The CPM Procedure..... 34

a) Forward pass 35

b) Backward pass 36

3.3 Slack times/Floats – Critical Path..... 38

4. THE CRITICAL RESOURCE DIAGRAMMING 40

4.1 CRD Network Development..... 40

4.2 CRD Computations & Analysis..... 41

4.3 CRD Node Classifications..... 42

5. METHODOLOGY. 44

5.1 Description..... 44

5.2 Applications..... 45

6 CRD COMPUTER MODEL..... 48

6.1 System Platform..... 48

6.2 Model Overview..... 48

6.3 Model Description..... 49

6.3.1 Assumptions..... 49

6.3.2 The Scheduling Module..... 50

6.3.3 The User Interface..... 52

6.3.4 The User Guide..... 53

6.4 CRD Sample Program Run..... 58

6.4.1 Input Data..... 58

6.4.2 Output Data..... 61

6.5 Evaluation of the CRD Computer Model..... 63

7. CONCLUSION AND RECOMMENDATIONS..... 66

7.1 Conclusion..... 66

7.2 Recommendations for Future Research.....67

REFERENCES..... 68

APPENDICES..... 76

Appendix A: DATA FOR TEST PROBLEM 77

Appendix B: CRD OUTPUT FOR PROBLEM TEST 1..... 79

Appendix C: CRD OUTPUT FOR PROBLEM TEST 2..... 84

VITA..... 88

LIST OF TABLES

TABLE	PAGE
2.1 – Notations and Definitions.....	11
5.1 – Sample project data.....	46
5.2 – Tasks durations for each resource.....	46
5.3 – Activities durations under CRD.....	46
6.1 – Project illustration.....	59
6.2 – Tasks durations for each resource in the specific activity.....	59
6.3 – Calculated activities durations under CRD.....	59
6.4 – CRD Output for example problem.....	62
6.5 – Calculated values comparison.....	65

LIST OF FIGURES

FIGURE	PAGE
1.1 – Origins of CRD and applications	6
2.1– Solution methods for the RCPSP and its variants	25
3.1 – Node representations	28
3.2 – Arrow representation	28
3.3 – Dummy activity graphical representations	30
3.4 – Merge event	30
3.5 – Burst event	30
3.6 – Representation of a precedence diagram	31
3.7 – Finish-to-start relationship	32
3.8 – Dummy start and finish nodes in a network	34
3.9 – Graphical representation of the network node for the corresponding activity	36
4.1 – Basic Critical Resource Diagram	41
4.2 – CRD Network Analysis	42
5.1 – Project network	46
5.2 – CRD network illustrations	47
5.3 – CPM network computations	47
6.1 – Modules interactions in the CRD model	49
6.2 – The CRD model start window	54
6.3 – The CRD interface under the multiple activities mode	55
6.4 – The CRD interface under the single activity mode	56
6.5 – CRD program interactions	57

LIST OF ABBREVIATIONS

AOA	activity-on-arrow
AON	activity-on-node
a_r	Constant per period availability of resource type r
CC	Critical Chain
CPM	Critical Path Method
CR	Indicator of criticality
CRD	Critical Resource Diagramming
csv	Comma separated values file
d_j	Duration of job j in period
DRES	Dummy resource
EC_i	Earliest completion time for activity i
EF_j	Earliest finishing time of job j
EFT	Earliest finish time
ES_i	Earliest start time for activity i
ES_j	Earliest starting time of job j
EST	Earliest start time
FS	Free slack
GA	Genetic algorithm
GUI	Graphical user interface
Html	Hyper text markup language
i	Activity identification
IF	Independent float
IS	Interfering slack
J	Set of all jobs; J = {1,……, n}
j	Index for the job; j = 1,….., n
LC_i	Latest allowable completion time for activity I
LFj	Latest finishing time of job j
LS_i	Latest allowable starting time for activity I

LIST OF ABBREVIATIONS (continued)

LST	Latest start time
LST	Latest finish time
m	Number of renewable resource types
n	Number of jobs
P(*i*)	Set of all the immediate predecessors of activity i
PDM	Precedence diagramming method
PERT	Project Evaluation Review Technique
Pj	Set of job which immediately precede / follow job j
PN	Precedence network
R	Set of all renewable resources types; R = {1,…, m}
r	Index for the renewable resource types; r =1,…., m
RAD	Rapid application development
RCPS	Resource Constrained Project Scheduling
RCS	Resource Constraint Scheduling
RES_j	Resource type j
ROT	Resource Over Time
R-R	Resource relationship
S(*j*)	Set of immediate successors of activity *j*
SA	Simulated Annealing
SM	Scheduling module
t	Index for periods; t = 1,….,T
T	End of the planning horizon
t_i	Duration of activity i
TOC	Theory of Constraints
TS	Tabu search
TS_i	Total slack of an activity i
UI	User interface
u_{jr}	Per period resource usage of resource type r by job j
VB	Visual basic 6.0

CHAPTER 1

INTRODUCTION

1. 1 Historical Perspective on Project Scheduling

In general, the scheduling of projects is concerned with the development of timetables, i.e., the establishment of periods during which the jobs required to complete the project will be executed. The most widely used scheduling approaches are based on creating a network that graphically depicts the relationships between the jobs of the project. Based on the developed network, the project scheduling is performed by applying forward pass and backward pass procedures. This results in earliest and latest starting and finishing times for the jobs. In addition, the amount of float or slack time associated with each activity, i.e., the amount of time a task may be prolonged or delayed without increasing the end of the project, can be calculated. The jobs with no slack are considered to be critical and require particular attention from the project manager. The corresponding computations are often referred to as CPM (critical path method) analysis.

Since their creation, the CPM models have been extended and improved to meet continuously changing requirements. Hartmann stated that the most significant improvement is the integration of resources [Hartmann 1999]. The consideration of resources in the scheduling process opens the road to a new field, the Resource-Constrained Project Scheduling (RCPS) problem.

In the RCPS problem, the project tasks are to be allocated based on precedence and resource constraints while taking into account a management objective, for example the minimization of the project's duration. A description of the Resource-Constrained Project Scheduling problem is presented in chapter 2.

1. 2 Resource Scheduling

Resource scheduling explicitly and systematically incorporates decisions about the capacity into the scheduling process. Gordon and Tulip (1997) stated that when using a network for the study or management of a project, or projects, there are five basic approaches available for modeling the resources involved in the project [Gordon et Tulip 1997]. These techniques are not mutually exclusive, depending on the experience of the project manager, they can be used alone or in combination with each other.

a) *Aggregation*- the simplest procedure for scheduling resources is probably aggregation. It is a method of determining the total number of resource units required on a unit time basis throughout the life of the project. The most frequently used unit of time is the day.

b) *Cumulation*- this approach provides a running accumulation of the resource requirements during the life of the project. The input to this process is the output of the aggregation calculation. In cumulation, the aggregation results are accumulated on a running basis. The results give to project managers the total resource usage that can be expected from the start to any time during the project execution. The 'traditional' use for this method has been when working with costs, which are in fact a particular form of resource. As reports of progress are obtained, the manager can quickly determine if expenses are in the bounds of reasonableness by checking that the figure lies within the envelope of the cumulation.

c) *Allocation*- the goal of this method is to provide a feasible schedule for the completion of the project within the management constraints. There are two different approaches to this problem; the serial and parallel allocation methods.

- Serial allocation is the fastest and simplest form of resource allocation. There are two distinct phases in the method:

Sequencing: all the activities to be scheduled are sorted into an ordered list.

Scheduling: the activities are then scheduled in the structured sequence determined in the previous phase. The plan is produced by calculating a schedule start time and a schedule finish time for each activity in turn. Each activity is considered alone and the start and finish dates are determined, by taking into account only the schedule completed at that time, the network constraints and the resource limits.

- Parallel allocation. The most obvious difference between this method and the serial allocation is that potentially all activities are available at every cycle of the project. This process has basically three steps, which keep cycling around until the schedule is complete:

- The first step is to select the time frame to be scheduled. Normally this is a single time unit and moves sequentially through the duration of the project from the beginning to the end.
- The second step is to select the set of activities, from those not scheduled so far, that could be in progress during this time frame and whose predecessors have been scheduled completely. This will include partially scheduled activities as well as those that have been delayed from previous cycles and some that would be expected from the time analysis.
- The last step is to scan through the set for the most important activity for which resources are available, schedule this activity for this time slot and continue cycling this step until all resources available in the slot have been exhausted, or the set is empty or no more activities in this particular set can be scheduled.

d) *Smoothing*- with this method, the objective is to produce a feasible schedule within the time constraints with a uniform level of resource requirement or planned usage. The smoothing procedure looks at the total project and works within the time

frames of the project that have been set by the user. There are four steps in the cycle, which repeat until all activities have been scheduled:

1. Schedule any critical activities.
2. Find the most important activity yet to be scheduled.
3. Find the best place to schedule this activity, and do so.
4. Adjust the early and late times of the unscheduled activities to take account of this newly scheduled activity.

e) *Leveling*- resource leveling refers to the process of reducing the period-to-period fluctuation in a schedule. If resource fluctuations are beyond acceptable limits, actions are taken to move activities or resources around in order to level out the resource-loading graph. Proper resource planning will facilitate a reasonably stable level of the work force. Acceptable resource leveling is typically achieved at the expense of longer project duration or higher project cost. It should be noted that not all of the resource fluctuations in a loading graph can be eliminated. Resource leveling attempts to minimize fluctuations in resource loading by shifting activities within their available slacks.

The techniques discussed later in this report, the Critical Resources Diagramming (CRD) method and the Critical Path Method (CPM), utilize the *allocation* procedure to generate a schedule for a defined project. More explicitly, they apply the serial allocation procedure to assign the required resources to a specific activity during the scheduling of a project.

1.3 The Statement of the Research

Project scheduling is the time-phased sequencing of activities subject to precedence relationships, time constraints, and resource limitations to accomplish specific objectives [Badiru and Pulat 1995]. The two main techniques of project management useful in the basic managerial functions of planning, scheduling, and control

are Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT).

1.3.1 Critical Path Method (CPM)

Initiated by DuPont Company along with Remington Rand in 1957, Critical Path Method (CPM) has become widely used, especially in the construction and process industries (Kerzner 2003). CPM is basically concerned with minimizing the duration of a project. The amounts of time needed to complete various aspects of the project are assumed to be known with certainty when applying the CPM method. Also the relationship between the amount of resources employed and the time needed to complete the project is assumed to be known. A description of the Critical Path Method is presented in chapter III.

1.3.2 Project Evaluation and Review Technique (PERT)

By 1959, Program Evaluation and Review Technique (PERT) was developed and used in cases where the techniques of Taylor and Gantt were inapplicable. In 1958, with the aid of the management-consulting firm of Booz, Allen, and Hamilton, PERT was introduced by the Special Projects Office of the U.S. Navy, on its Polaris Weapon System. PERT takes into account the uncertainties in the activity duration time estimates. It assumes that the activities and their network relationships have been well defined, but it allows for uncertainties in activity times. Not only is an estimate made of the most probable time required to complete the activity (denoted by "m") for each activity in the network, but some measure of uncertainty is also noted for this estimate. The other two time estimates for the PERT procedure are the pessimistic estimate, denoted by "b" and the optimistic estimate, denoted by "a". These three time estimates are given by the person who would be most qualified to know – an engineer, supervisor, or worker.

A common characteristic of the methods presented above is their focus on activities during the scheduling process. Badiru, in 1993, introduced a method called

Critical Resource Diagramming, which, focuses on resources before scheduling the activities.

1.3.3 Critical Resource Diagramming (CRD)

In his study, Badiru proposes the use of a new tool called Critical Resource Diagram (CRD) for a better resource management [Badiru 1993]. CRD is a simple extension to the CPM technique developed for resource management purposes. Unlike activity networks, the CRD uses nodes to represent each resource unit. Also, a resource unit may appear more than once in a CRD network, specifying all different tasks for which a particular unit is assigned to. In the diagram, a node represents a resource unit and has its own duration originated from the resource owner (activity). A node also has links to other nodes based on the technical relationship between activities. Similar to CPM, the same backward and forward computations may be performed with CRD. Figure 1.1 presents the evolution from the critical path method to the critical resource diagramming and domains in which the use of the CRD method can be of great interest.

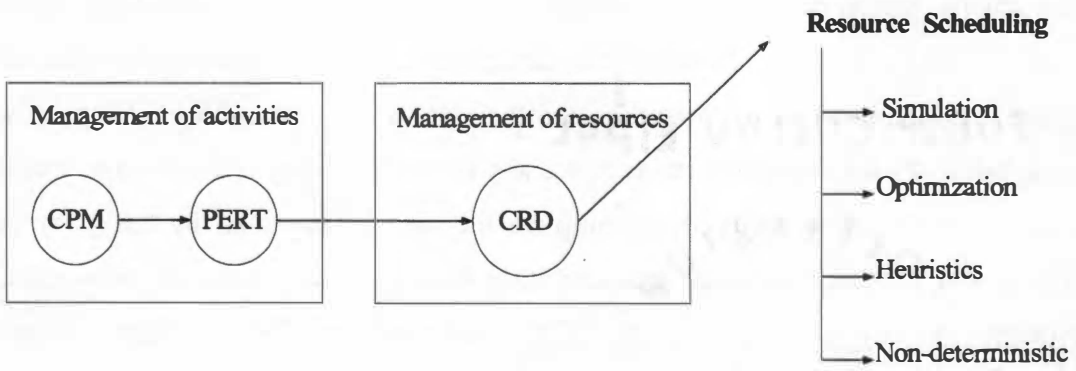

Figure 1.1 Origins of CRD and applications

1.3.4 Objectives of the Research

As stated earlier, the consideration of the resource limitations in project analysis leads to the Resource-Constrained Project Scheduling Problem (RCPSP). Different methods, for example the CPM method, have been used to solve specific cases of the RCPSP. The CRD method can also be used in solving a RCPSP case.

This present study explores that advantage of CRD. The purposes are to develop a methodology, based on CRD, which allows its use in specific case of the RCPSP, to implement the developed technique using a computer model, to conduct test to validate the CRD computer model, and then to investigate the advantages and disadvantages of the introduced method.

The CRD computer model will be implemented using the Visual Basic 6.0 language. The inputs to the simulation model are:

1. Number of activities
2. Number of resources
3. Resource requirements for each activity
4. The time estimate for a resource to perform a specific task during an activity
5. Activity precedence relationships

Using the previous inputs, the program attempts to schedule the project using the resources and allocating them to each activity without violating the precedence relationship between activities and the resource constraints.

1.4 Thesis Organization

The following chapter presents the basics concepts of the RCPSP problem and reviews the methods and applications used to attempt to solve the RCPSP problem problems.

Chapter 3 discusses the fundamentals of the project network analysis; a presentation of the CPM method is also included. Chapter 4 concentrates on the CRD technique. Chapter 5 focuses on the methodology developed in this research. Chapter 6 gives a description of the implemented computer application and the results of the evaluation. Chapter 7 is the concluding chapter and also contains ideas for further studies using this research.

CHAPTER 2

LITERATURE REVIEW

Project scheduling has been and still is an area of intensive research over the years. This is due to the variety of project scheduling problems that arise in the professional world. Despite the wide range of scheduling problems, close looks at their characteristics lead to classifying them into two categories: the stochastic and deterministic scheduling problems [Herroelen and al. 1999].

The *Stochastic Scheduling Problem* refers to cases where the characteristics are stochastic. In most cases, the stochastic characteristics apply to the activity durations and the amount of the cash flows. Other specifications such as the resource availabilities and requirements can also have stochastic classification.

The *Deterministic Scheduling Problem* includes cases where the different parameters such as activity duration, precedence relations, and resource requirements are assumed to be deterministic and known in advance.

This group can further be subdivided into *Scheduling in the absence of resource constraints* and *Scheduling under resource constraints*. The first group is concerned with the time or cost analysis of activity networks; classical methods like PERT or CPM can be used to solve them. In the Project scheduling under resource constraints category, different kinds of problems have been classified due to the variety of resource constraints and/or project characteristics, for example the execution mode of the activities within the project. The most notable types of problems are [Herroelen and al. 1999]:

- The resource constrained project scheduling problem and its variants.
- The time/cost trade-off problem
- The discrete time/resource trade-off problem
- The resource leveling problem
- The resource constrained project scheduling with discounted cash flows

2.1 The Resource-Constrained Project Scheduling Problem

Resource constrained project scheduling problem (RCPSP) involves assigning jobs or tasks to a resource or set of resources with limited capacity in order to meet some predefined objective. Many different objectives are possible and these depend on the goals of the decision maker. The most common goal is to find the minimum makespan, i.e, the minimum time to complete the entire project.

2.1.1 The Problem

In its most general form, the resource-constrained scheduling problem can be defined as follows:

> Given a set of activities that must be executed in one of several modes, a set of resources with which to perform the activities under the selected mode, a set of constraints that must be satisfied, along with a set of objectives with which to judge a schedule's performance. What is the best way to assign the resources to the activities at specific times such that all of the constraints are satisfied and the best objective measures are produced?

Through the years, different mathematical formulations have been used to represent the RCPSP. The mathematical techniques used are based on integer programming that formulates the problem using 0 and 1 indicator variables. The variables indicate if an activity is scheduled within specific time periods [Badiru 1993].

The following is a presentation of two mathematical formulations of the resource-constrained scheduling problem. Table 2.1 shows a summary of the notations and terms used.

- Formulation 1

One of the first mathematical formulations explicitly developed for the resource constrained scheduling problem has been by Pritsker and al in 1969 [Klein 2000].

Table 2.1 Notations and Definitions (Klein 2000)

Notations	Definitions
n	Number of jobs
J	Set of all jobs; J = {1,……, n}
j	Index for the job; j = 1,….., n
d_j	Duration of job j in period
Pj	Set of job which immediately precede / follow job j
$\overline{T}$	End of the planning horizon
t	Index for periods; t = 1,…., $\overline{T}$
ES_j	Earliest starting time of job j
EF_j	Earliest finishing time of job j
LFj	Latest finishing time of job j
m	Number of renewable resource types
R	Set of all renewable resources types; R = {1,…, m}
r	Index for the renewable resource types; r =1,…., m
a_r	Constant per period availability of resource type r
u_{jr}	Per period resource usage of resource type r by job j

In this formulation, solutions are defined as follows:

$$x_{jt} = \left\{ \begin{cases} 1, \text{ if job j is finishedat the end of period t} \\ 0, \text{ otherwise} \end{cases} \right\} \text{for } j \in J \text{ and } t \in \left[EF_j, LF_j\right]$$

For a given solution vector of an RCPSP instance, the period SF_j in which a job j is scheduled to be finished is determined by:

$$SF_j = \sum_{t=EF_j}^{LF_j} t \bullet x_{jt}$$

Based on the above parameters and variables, the following model was developed

$$\text{Minimize} \quad CT(x) = \sum_{t=EF_n}^{LF_n} t * x_{nt} \qquad \text{(a)}$$

Subject to

$$\sum_{t=EF_j}^{LF_j} x_{jt} = 1 \qquad \text{for} \quad j \in J \qquad \text{(b)}$$

$$\sum_{t=EF_j}^{LF_j} (t - d_j) * x_{jt} - \sum_{t=EF_i}^{LF_i} t * x_{it} \geq 0 \quad \text{for } j \in J \text{ and } i \in P_j \qquad \text{(c)}$$

$$\sum_{j \in E(T)} u_{jr} \cdot \sum_{q=\max\{t, EF_j\}}^{\min\{t+d_j-1,\, LF_j\}} x_{jq} \leq a_r \quad \text{for } r \in R \text{ and } t \in [1 \ldots\ldots, \overline{T}] \qquad \text{(d)}$$

$$x_{jt} \in \{0,1\} \qquad \text{for } j \in J \text{ and } t \in [EF_j, LF_j] \qquad \text{(e)}$$

The objective function (a) minimizes the completion time of the dummy end job *n* and therefore, the project's makespan. The assignment constraint (b) together with the integrality constraint (e) guarantee that exactly one finishing period is assigned to each job j. Constraint (c) corresponds to the precedence restrictions and constraint (d) represents the resources requirements for each period $t \in [1,\ldots\ldots,\overline{T}]$ and each resource type $r \in R$

Formulation 2

The subsequent formulation was originally developed by Kaplan [Kaplan 1988] to model a resource-constrained project scheduling problem with preemption and by Neumann and Morlock [Neumann and al 1993] for the resource-leveling problem [Klein 2000]. It was then adapted to the RCPSP. Similar to the previous one, this formulation is also based on defining 0 - 1 variables, which indicate whether a job j is active in period t or not:

$$x_{jt} = \left\{ \begin{array}{l} 1, \text{ if job j is finished at the end of period t} \\ 0, \text{ otherwise} \end{array} \right\} \text{ for } j \in J \text{ and } t \in [EF_j + 1, \ldots, LF_j]$$

This formulation requires job duration larger than zero because otherwise the fulfillment of the precedence constraints cannot be verified. Using the binary variables x_{jt}, the resource constraints project scheduling problem is formulated as follows:

$$\text{Minimize} \quad CT(x) = \sum_{t=ES_n+1}^{LF_n} t * x_{nt} \tag{1}$$

Subject to

$$\sum_{t=ES_j+1}^{LF_j} x_{jt} = d_j \qquad \text{for} \quad j \in J \tag{2}$$

$$d_j \cdot (x_{jt} - x_{j,t+1}) - \sum_{q=ES_j+1}^{t} x_{jt} \leq 0 \qquad \text{for } j \in J \text{ and } t \in [ES_j + 1, \ldots, LF_j - 1] \tag{3}$$

$$d_j \cdot x_{jt} - \sum_{q=ES_j+1}^{t-1} x_{jt} \leq 0 \qquad \text{for } j \in J, i \in P_j, t \in [ES_j + 1, \ldots, LF_j - 1] \tag{4}$$

$$\sum_{j \in E(t)} u_{jr} \cdot x_{jt} \leq a_r \qquad \text{for } r \in R \text{ and } t \in [1, \ldots, \overline{T}] \tag{5}$$

$$x_{jt} \in \{0,1\} \qquad \text{for } j \in J \text{ and } t \in [ES_j + 1, \ldots, LF_j] \tag{6}$$

The objective function (1) minimizes the project completion time, because the given sum yields the smallest possible value if the dummy end job n is completed as early as possible. The duration constraint (2) ensures that each job is executed for d_j periods.

To guarantee that the processing of each job is not interrupted, the non-preemption constraint (3) is introduced. A job is only allowed to be complete in a period t but not in the subsequent period t +1, if it is terminated at the end of period t. This is the case when it has been processed for d_j periods. The precedence constraint (4) follows the same idea. A job j must not be started before all its predecessors i $\in P_j$ have been processed completely, i.e., for d_i periods. The restriction (5) represents the resources constraints.

Others formulations have been presented by Klein (2000), Talbot (1982), Yang, Geunes and O'Brien (2001), Kolisch and Hartmann (1999), Cheng and Gen (1998), Selle and Zimmermann (2002), Brucker, Drexl, Mohring, Neumann and Pesch (1999), Hartmann (1999), and Böttcher, Drexl, Kolish, and Salewski (1999)

2.1.2 Characteristics of the RCPSP

a) Activities

Activities are defined using specific parameters such as duration, cost, and resource consumption. Any task may require a single resource or a set of resources, and the resource usage may vary over the duration of the task.

A job may have multiple execution modes. Each mode reflects a feasible way to combine cost, duration, and resource consumption that allows a satisfying completion of the specific activity. For example, fifty construction workers may dig a mine within a month, where an excavator may perform the same work within a week. Some tasks, once begun, may not be stopped, nor their mode switched, until the activity is complete. Alternatively, some activities may be aborted at any time, possibly with some corresponding cost. Interruption modes may depend on the resources that are applied to the task. Some tasks may be interrupted, but the resources they use cannot be used elsewhere until the work is finished. Other tasks may be interrupted, the resources reassigned, then any resource reapplied when the task is resumed (preemption).

The RCPSP assumes that an activity can only be executed in a single mode. A single mode is defined by a fixed duration and fixed resource requirements. Once started in one of its modes, an activity must be completed in that mode; mode changes and preemption are not allowed [Hartmann 1999].

b) Resources

Resources used by a specific project are classified in four different categories, namely renewable, nonrenewable, doubly constrained and partially renewable resources. The first three categories have been defined by Slowinski, Blazewicz and Weglarz [Slowinski and al 1986], the last category was defined by Böttcher [Böttcher and al 1999].

Renewable resource includes resources in which capacities are limited on each time period of the project duration. Examples of renewable resources include labor and many types of equipment. Non-renewable resources concern resources that are limited for the whole project duration. Examples of non-renewable resources are capital and raw materials. Resources are doubly constrained when their capacity is limited on each time period and on the whole project length; capital can be considered as an example in that case also. The partially renewable resource category applies to resources in which capacities are limited on a subset of the project duration [Drezet and Tacquard 2003]. In some cases, renewable resources may become non-renewable resources, in others, non-renewable resources may be considered renewable.

The basic RCPSP assumes the use of renewable resources [Hartmann 1999], the other types of resources are used when dealing with variants of the resource constrained scheduling problem.

c) Constraints and Objectives

Constraints and objectives are defined during the formulation of the problem. Constraints define the feasibility of a schedule. Objectives characterize the optimality of a schedule. Project scheduling problems typically specify time-based objectives such as

the minimization of project duration as the primary objective. The nature of the project objectives can also be resource-based, for instance, resource investment or resource leveling, financially related objectives such as maximize the project net present value or they can also be quality oriented objectives which include projects where the goal is to minimize the rework time or the rework cost. However, in many cases, real-world problems are subject to multiple, often conflicting, objectives.

Similar to the objectives, constraints appear in many forms. Precedence constraints define the order in which tasks can be performed. Temporal constraints limit the times during which resources may be used and/or tasks may be executed. Activity constraints define the method used to perform the activities.

2.1.3 Instances of the RCPSP

Throughout the literature, many variations of the resource constrained project scheduling problem have been examined. These variations can be classified using the different above-defined classes for each of the components of the RCPS problem. The following characteristics can be used for a clear definition of the problem studied [Cheng and Gen 1998]

Project number. The number of simultaneously scheduled projects

- Single project
- Multiple projects

Execution mode. The execution modes of activities

- Single mode
- Multiple modes

Resource utilization. The characteristics of the employed resources

- Renewable
- Non-renewable
- Doubly-constrained
- Partially renewable

Objectives. The nature of the objective

- Minimize the project duration
- Minimize the project cost
- Maximize the project net present value

Interruption mode. The interruption of activity execution

- Non-preemptive case
- Preemptive case

The characterization of the RCPSP case studied is important because of the RCPSP complexity. So far, a general method able to solve any case of the resource constrained project scheduling problem has not been developed only solutions for specific cases have been presented. Thus it is necessary when dealing with RCPSP to identify the specifics of the problem studied.

The type of resource-constrained project scheduling problems studied in this research can be defined as follows:

- A single project is considered
- Each project consists of a number of activities with known duration
- Precedence constraints between activities
- Resources are available in limited quantities but renewable from period to period
- Activities cannot be interrupted (non-preemptive case)

2. 2 Different Solutions to RCPSP

Variations of the resource-constrained scheduling problem have been proposed, implemented, and evaluated for over fifty years. The solution methods form two distinct classes: exact methods and heuristic methods. Exact methods are guaranteed to find a solution if it exists, and usually provide some indication if no solution can be found. Heuristic solutions may have no such guarantee, but typically assure some degree of optimality in their solutions.

Early attempts to solve the resource-scheduling problem used mathematical models such as linear programming, integer programming, and dynamic programming. While very powerful when used on small-scale problems, their efficiency usually decreases for large-scale problems due to a phenomenon called "combinatorial explosion" [Senouci and Adeli 2001]. Special algorithms have been developed, to overcome the difficulties associated with the combinatorial explosion, some of which are the Branch and Bound Approach [Patterson and Roth 1976] and the Implicit and the Bounded Enumeration Method [Davis and Heidorn 1971; Talbot and Patterson 1978; Demeulemeester and Herroelen 1997].

Another alternative to improve the computational efficiency come with the use of heuristic methods [Wiest and Levy 1977; Boctor 1990]. The variety of network structures and resources made it so that no single heuristic method can always give the best available solution for all type of resource scheduling problems, thus a large number of heuristic algorithms have been developed and tested [Alvarez-Valdes and Tamarit 1989].

Heuristics rules generally give satisfying results and are widely used in practice because of their simple format and efficiency in application; however, optimal solutions are not guaranteed based upon the heuristic method. Recently, research has been directed towards computational optimization techniques such as genetic algorithms and simulated annealing. [Cheng and Gen 1998, Shixin and Mengguang 2000]

2.2.1 Exact Solution Methods

Exact methods are guaranteed to find the optimal solution, but typically become impractical when faced with problems of any significant size or large sets of constraints.

a) CPM - PERT

Essentially, there are six steps, which are common to both the techniques. The procedure is listed below:

- Define the project and all of its significant activities or tasks. The project (made up of several tasks) should have only a single start activity and a single finish activity.
- Develop the relationships among the activities. Decide which activities must precede and which must follow others.
- Draw the "Network" connecting all the activities. Each activity should have unique event numbers. Dummy arrows are used where required to avoid giving the same numbering to two activities.
- Assign time and/or cost estimates to each activity
- Compute the longest time path through the network. This is called the critical path.
- Use the Network to help plan, schedule, monitor and control the project.

The Key Concept used by CPM/PERT is that a small set of activities, which make up the longest path through the activity network, control the entire project. If these "critical" activities could be identified and assigned to responsible persons, the resources management could be optimally done by concentrating on the few activities that determine the outcome of the entire project.

Non-critical activities can be replanned, rescheduled and their resources can be reallocated flexibly, without affecting the whole project. The CPM/PERT provides the resource-unconstrained schedule for a set of precedence-constrained activities with deterministic durations. It gives the shortest possible makespan assuming infinite resources. Although useful for obtaining a rough idea of the difficulty of executing a plan, the critical path method does not consider temporal or resource constraints.

b) Mathematical Programming

Many scheduling problems have been formulated using mathematical programming. The general purpose of linear programming, integer programming and dynamic programming is to find the single optimal answer [East 2001].

Samuel Gorenstein [Gorenstein 1972] presented a mixed-integer programming method. In his approach, he developed an algorithm that uses a partial enumeration procedure and a maximum-flow computation as a check for feasibility with respect to available resources. The technique, with the objective of minimize the project duration, is said to have the benefit of eliminating the need to consider individual time period over the project horizon.

Another approach, which uses integer programming, was offered by Talbot and Patterson [Talbot and Patterson 1978]. The algorithm developed, allocates limited resources to competing activities of a project such that the project duration is minimized. The procedure consists of a systematic evaluation of all finish times for each activity in the project. In order to limit the number of explicit evaluations, the authors used a method called a network cut, which removes from consideration the evaluation of finish times, which cannot lead to a reduced project completion time. Results, reported by the authors, indicated that the method is a reliable optimization technique for projects consisting of up to 50 jobs with three different resource types.

c) Branch-and-Bound Method

Early attempts at using integer programming to solve the exact version of the resource constrained project-scheduling problem were unsuccessful [Demeulemeester and Herroelen 1992]. In consequence, numerous enumerative (branch-and-bound) methods for solving certain variants of the problem optimally were developed. Variations of branch-and-bound solution methods were first proposed in the 1960s [Lawler and Wood 1966][Johnson 1967] Since then, varieties of branch-and-bound methods that are based on different concepts have been presented.

Patterson et al proposed an algorithm guided by the precedence tree concept [Patterson and al 1989][Brucker and al 1999].

Another concept used to develop a branch-and-bound approach is the delay alternatives. Demeulemeester and Herroelen presented a branch-and-bound technique using the delay alternatives concept [Demeulemeester and Herroelen 1992] [Hartmann 1999].

Other branch-and-bound approaches have been presented by Balas (1970), Stinson and al (1978), Bell and Park (1990), Christofides and al (1987), Sprecher (2000), Talbot and Patterson (1978), Mingozzi and al (1998), Hartmann (1999), and Böttcher and al (1999).

Recently, simplify and accelerate techniques, called bounding rules, have been created to reduce the effort in the use of the enumeration techniques [Sprecher and al.1995]. Some of these rules are: the extended and simplified single enumeration rule, the local left-shift rule, the extended global left-shift rule, the contraction rule, and the set-based dominance rule.

As noted by Sprecher and Drexl, enumerative methods cannot solve large problems; the tree is simply too big. Although significant progress has been made in the pruning techniques, branch-and-bound methods are still limited to less than one hundred activities or even fewer in the multi-mode cases, and they still require special heuristics to accommodate variations in resource constraint formulations. [Sprecher and Drexl 1996] [Wall 1996]

d) Critical Chain

The Critical Chain (CC) methodology for project management (CCPM), which was presented by Dr Eliyahu M. Goldratt (1997) in his well-known book Critical Chain, can be define as the direct application of the theory of constraints (TOC) to project management [Herroelen, Leus and Demeulemeester 2002]. In their publication (2004), Cohen, Mandelbaum and Shtub presented the steps of the Critical Cain approach as follows:

- Reduce activity durations by eliminating safety margins.

- Identify the critical chain
- Create a project buffer
- Create feeding buffers
- Control-Buffer monitoring

Further discussions on critical chain have been presented by Leach (1999), Herroelen, Leus and Demeulemeester (2002), Raz, Barnes and Dvir (2003), and Cohen, Mandelbaum and Shtub (2004).

2.2.2 Heuristic Solution Methods

Results, published in the related literature, have shown that the computational complexity of exact methods becomes very unaffordable for real-world size projects. As a consequence, the development of heuristics techniques, which required far less computational time and memory space, has been the center of interest for many researchers. Since the year 1963, that saw the development of the first heuristic approach by Kelley [Kelley 1963, Hartmann 1999], a large number of different heuristic algorithms have been suggested in the project scheduling area.

Most of the heuristics, presented for solving the resource-constrained project scheduling problem, have been classified into two categories, namely constructive heuristics and improvement heuristics. [Klein 2000, Demeulemeester and Herroelen 2002]

Constructive heuristics start from an empty schedule and perform a set of operations that lead to the scheduling of the activities, one by one, until one feasible schedule is obtained. To that purpose, the activities are habitually ranked using priority rules, which determine the order in which the activities are scheduled.

Improvement heuristics, on the other hand, begin with a feasible schedule, which has already been defined using some constructive method, and then successively perform operations that transform a schedule into an improved schedule.

a) Constructive Heuristics

Constructive heuristics generally consists of two main components, the scheduling scheme and the priority rule. They combine priority rules and schedule generation schemes to create a specific algorithm, which will develop a feasible schedule.

The scheduling scheme defines the approach in which a feasible schedule is created by assigning starting times to the different tasks. Two basic scheduling schemes have been identified: the serial scheduling scheme and the parallel scheduling scheme.

The priority rule determines the activity that is chosen next during the scheduling process. Application of the priority rule results in a priority list in which the activities are ordered in decreasing priority without violating the precedence constraint. Different categories of priority rules have been presented:

- Activity based priority rules
- Network based priority rules
- Critical path based priority rules.
- Resource based priority rules
- Composite priority rules

b) Improvement Heuristics

Improvement heuristics, as indicated by their name, are based on improving an already existing solution. They use a current feasible schedule, defined using constructive methods, and then successively perform operations that transform the cited schedule into an improved schedule according to the objectives of the project. The operation is then performed until no more improvement is possible.

The most common approaches are the *descent approaches*; they develop a series of solutions, under the assumption that each solution improves upon the previous one. The computations are executed as long as an improved solution can be found in the neighborhood of the last solution.

Unfortunately, the use of the above-described methods have been limited due to the possibility of cycling; solutions may be visited more than once and other solutions may not, during the search process [Klein 1999]. Thus, new improvement heuristics, called metaheuristic, have been developed to overcome this limitation.

The most common metaheuristic methods are simulated annealing (SA), tabu search (TS) and genetic algorithm (GA).

Studies on SA, TS and GA approaches have been presented by Aarts et al (1988), Lee and Kim (1996), Palmer (1994), Boctor (1996), Bouleimen and Lecocq (1998) and Cho and Kim (1997). Pinson and al (1994), Lee and Kim (1996), Baar and al (1998), Hartmann (1999), Lee and Kim (1996), Leon and Balakrishan (1995) and Kohlmorgen and al (1999).

Relative to the project-scheduling problem, the CRD method may be classified in the category of exact methods for project scheduling under resources constraints. The CRD uses the available resources to schedule the project.

Figure 2.1 presents a summary of the different solutions of the RCPSP along with the type of problem in which they can be applied. Furthermore, another categorization is introduced. We differentiate between prescriptive solutions and descriptive solutions.

Prescriptive solutions consider the methods that generate an optimal schedule. In general no improvement is required.

Descriptive solutions regroup the methods that generate a basic schedule, which can be improved

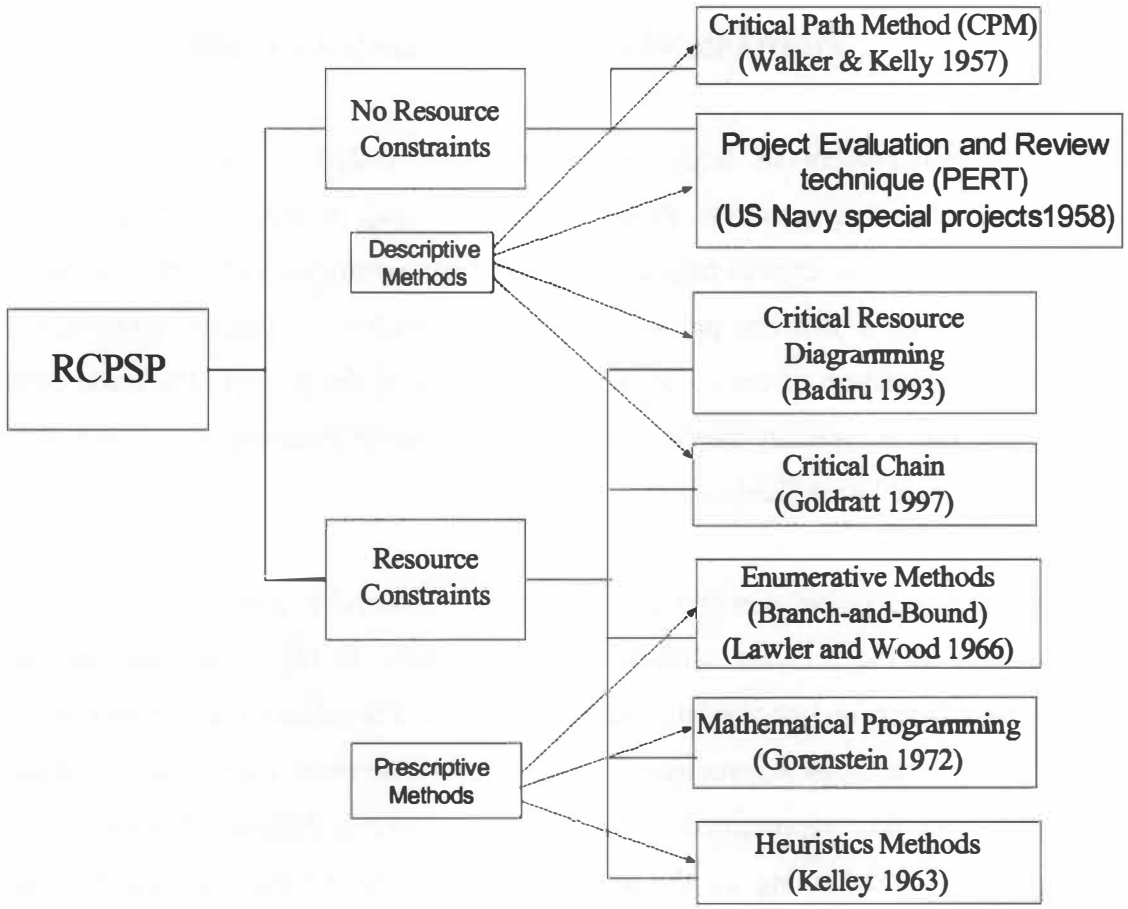

Figure 2.1 Solution methods for the RCPSP and its variants

CHAPTER 3

FUNDAMENTALS OF NETWORK ANALYSIS

The management of large projects requires analytical tools for scheduling activities and allocating resources. Project network analysis provides the manager with a set of tools, such as the critical path method (CPM) or the project evaluation and review technique (PERT), which has proven to be very proficient. A project network is the graphical representation of the contents and objectives of the project. The basic project network analysis is typically implemented in three phases: Planning phase, Scheduling Phase, and Control Phase [Badiru 1993]:

- Planning phase sometimes referred to as activity planning, involves the determination of the relevant activities for the project. In this phase, the necessary activities and their precedence relationship are defined. Precedence requirements may be defined using technological, procedural, or imposed constraints. Using this information, the activities are then represented in the form of a network diagram. The two popular models for network drawing are the *activity-on-arrow* (AOA) and the *activity-on-node* (AON) conventions. In the AOA approach, arrows are used to represent activities, while nodes represent starting and ending points of activities. In the AON approach, nodes represent activities, while arrows represent precedence relationship; the Critical Resource Diagram uses the AON convention. Time, cost, and resource requirement estimates are developed for each activity during the planning phase.

- Scheduling phase corresponds to a computational phase. During this phase, the scheduling is performed using forward pass and backward pass computational procedures. These computations give the earliest and latest starting and finishing times for each activity. The amount of slack or float associated with each activity is determined. The activity path with the minimum slack in the network is used to determine the critical

activities. This path also determines the duration of the project. Resource allocation, and time-cost trade-offs are other functions performed during scheduling.

- Control phase involves tracking the progress of a project on the basis of the network schedule and taking corrective actions when needed. An evaluation of actual performance versus expected performance determines deficiencies in the project evolution.

3.1 Project Network Analysis

Historically, network analysis procedures originated from the traditional Gant chart, or bar chart, developed by Henry L. Gantt during World War I [Bedworth 1982]. As mentioned previously, the two popular systems of networking are the activity-on-arrow (AOA) and the activity-on-node (AON). Many terms and symbols are recurrent to all types of network analysis. The following part gives a description of the major terms and symbols used. More detailed presentation on network components are given by Moder and Phillips (1970), Conway (1967), Elmaghraby (1977), and Badiru and Pulat (1995)

1. Node

A node is any closed geometric figure such as a circle, square or diamond. Figure 3.1 illustrates different node representations

2. Arrow

An arrow is a line connecting two nodes and having an arrowhead at one end. Under the AOA convention, the tail of the arrow corresponds to the start of the activity and the head represent the completion of the activity. With the AON convention, the arrow implies that the activity at the tail of the arrow precedes the one at the head of the arrow. Figure 3.2 shows an arrow representation



Figure 3.1 Node representations



Figure 3.2 Arrow representation

3. Activity

An activity is the smallest self-contained unit of work used to define a portion of a project. In general, activities have the following characteristics: a definite duration, logic relationships to other activities in a project, use resources such as people, materials or facilities, have an associated cost and explicit start and end point. In the AOA system, an activity is represented by an arrow and under the AON convention, a node will represent the activity. The task the activity represents may be indicated by a short phrase or symbol inside the node or along the arrow. Based on the logic relationships between activities in a project, an activity can be listed as:

- Predecessor activity: An activity that must be completed or be partially completed before a specified activity can begin is called a predecessor activity.
- Successor activity: An activity that can only start after one or more other activities are completed or partially completed is called a successor activity. A successor activity is dependent on a predecessor activity.
- Descendent activity: a descendent activity is any activity that can start only after a specified activity has been completed or partially completed.
- Antecedent activity: an antecedent activity is any activity that must be completed or partially completed before a specified activity can be started.

4. Duration

Duration of an activity is the amount of time needed to complete an activity. Duration is normally expressed in working days. It may also be expressed in months, weeks or hours.

5. Dummy activity

A Dummy Activity is an activity of zero duration used to show a logical relationship in the network method. Dummy activities are used when logical relationships cannot be completely described with regular activity representation. Dummies are shown graphically as a dashed arrow or dashed node and do not require resources. A dummy is never required in the AON system but it may be included to maintain the logic of the network. Figure 3.3 points up the graphical representation of the dummy activity

6. Event

An event is a moment in the project duration representing the start or completion of one or more activities. There is no time duration associated with an event Graphically, an event is represented by a node and labeled with a unique number. A significant event is often called a milestone. Due to the activities relationship, an event can be further categorized into:

(i) Merge event: A merge event is an event at which the completion of two or more activities precedes the start of the following activity as shown in Figure 3.4. All the preceding activities have to be completed before the start of the merge event.

(ii) Burst event: A burst event is an event in which two or more activities commence after the completion of the preceding activity as shown in Figure 3.5. None of the activities immediately following the burst event can be started until the burst event is completed.



Figure 3.3 Dummy activity graphical representations

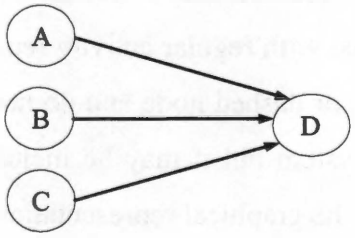

Figure 3.4 Merge event

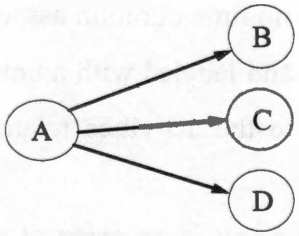

Figure 3.5 Burst event

7. Precedence Diagram

A precedence diagram is a graphical image describing the sequence of activities that compose a project. The activities are represented by nodes, which are then interconnected using arrows to illustrate the precedence relationship between activities based on precedence requirements. A precedence diagram is given in Figure 3.6. The number in each node represents the activity duration.

In establishing the precedence requirements or dependencies between activities, different type of constraints can be considered. A further understanding of those constraints leads to classify them into three basic categories [Badiru 1993, Kerzner 2003]:

- Technological constraints (i.e. hard logic). These are requirements caused by the technical relationships among activities. As an example, in a building construction, the activity *install the roof* is technically dependent upon the activity *erect the walls*.

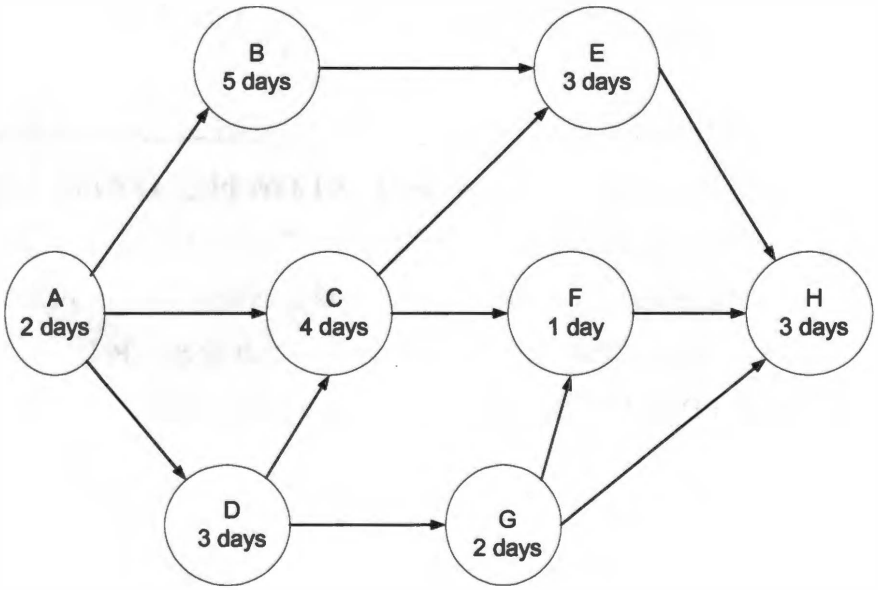

Figure 3.6 Representation of a precedence diagram

- Procedural constraints (i.e. soft logic). These are requirements determined by the policies and procedures under which the project is accomplished. These types of restrictions can usually be avoided.

- Imposed constraints. These are requirements that are beyond the control of the project manager. They can be classified as resource-imposed, state-imposed or environment-imposed. The most common constraint being the resource-imposed one. For example, resource shortage may require that one task be completed before another. The resource limitation is usually the most important part of developing a project schedule.

Based on the above constraints or dependencies between activities, there are four types of precedence relationships:

- Finish-to-start
- Start-to-start
- Finish-to-finish
- Start-to-finish

The finish-to-start relationship is the only type of precedence relationship allowed in typical PERT or CPM program. In a finish-to-start relationship; the start of the successor activity is constrained by the finish of the predecessor. Figure 3.7 gives an illustration of a finish-to-start relationship. The three other relationships are incorporated in an extension of PERT and CPM called precedence network (PN) or precedence diagramming method (PDM) [Kerzner 2003; Wiest and Levy 1977].



Figure 3.7 Finish-to-start relationship (Kerzner 2003)

In completing the diagram, the applications of the following rules have been found efficient in developing a clear and comprehensive model.

1. The network has only one initial node and one terminal node.
2. All nodes, except the terminal node, must have at least one following node.
3. All nodes, except the first node, must have at least one preceding node.

4. Node numbers must not be duplicated in a network.
5. Any two nodes may be directly connected by no more than one arrow.
6. No arrow must be left dangling. Every arrow must have a starting node and an ending node.
7. Arrows represent the logical precedence only. Neither the length of the arrow nor its "compass" direction has any significance. Although it is conventional to show the arrows directed from left to right.
8. The network must not have loops.

Generally, a project network will have a single initial and terminal node. This is not always the case. Due to the type of project, more than one initial or terminal node can occur. The AON convention eludes this difficulty by adding a dummy start or finish node to the network. This is shown in Figure 3.8.

Once the network representing the sequence of a project is constructed, concluding the planning phase, the next step is the scheduling phase. This phase involve the time analysis of the network using computational procedure.

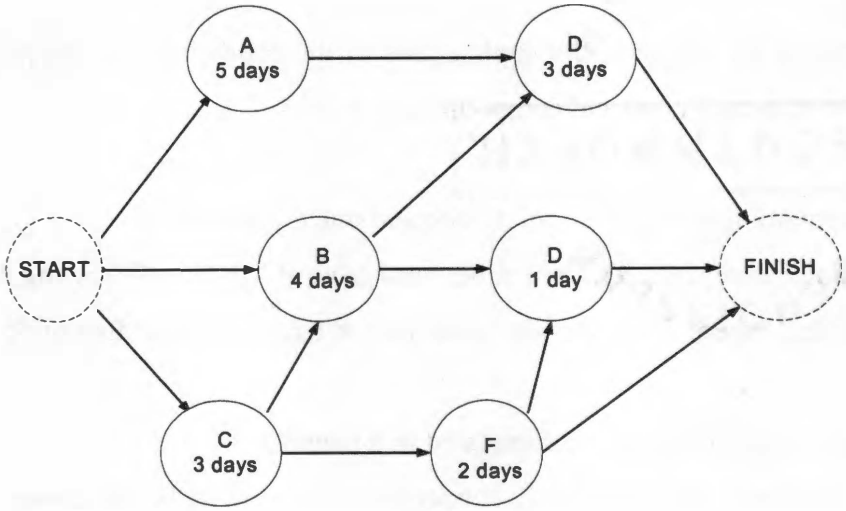

Figure 3.8 Dummy start and finish nodes in a network

3.2 The CPM Procedure

The CPM method is a scheduling algorithm, which was developed by M.R Walker of E.I. Dupont de Nemours & Co. and J.E.Kelly of Remington Rand, in 1957. The first test was made in 1958, when CPM was applied to the construction of a new chemical plant. In March 1959, the method was applied to a maintenance shutdown at the Dupont works in Louisville, Kentucky. Unproductive time was reduced from 125 to 93 hours.

The main objective in the CPM technique is to determine the "critical path". The critical path is the longest chain of project activities through the network, and its duration is equal to the minimum completion time of the project. A specific project can have more than one critical path through its network; all the critical paths have the same length. The determination of the "critical path" involves two computational procedures called forward pass and backward pass.

In conducting the forward pass, starting from the initial activity to the final one, the activity duration is used to determine the earliest start time and the earliest finish time for each activity in the network. The backward pass will help determine the latest starting

time and latest completion time for each activity. The following notations are used to represent the information about each project activity during the computation phases:

i: Activity identification

t_i: Duration of activity i

ES_i: Earliest start time for activity i

EC_i: Earliest completion time for activity i

LS_i: Latest allowable starting tine for activity i

LC_i: Latest allowable completion time for activity I

In the literature, different graphical representations are used to represent the above project activity's information on the corresponding network node. Figure 3.9 shows the graphical representation selected for this study.

***a)* Forward pass**

The forward pass helps determine the earliest start time (ES_i) and earliest completion time (EC_i) for each activity in the network. During the computations, the start time for an activity is assumed to be as early as possible, i.e., as soon as all of the activity's predecessors are completed. Thus the earliest start time for an activity is equal to the maximum EC_i of all of its predecessors. To initiate the calculations, an arbitrary earliest start time is assigned to the initial project activity. Usually, a value of zero is assigned to this initial start time; this will allow subsequent earliest start time to be interpreted as the project duration up to the point in question.

The computations process follows the steps presented below, starting from the initial project node to the end node:

Step 1. Assign an early start time of zero to the first node, noted node 1, representing the initial project activity in the network diagram.

$$ES_1 = 0$$

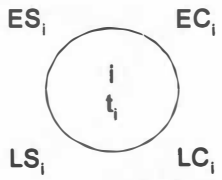

Figure 3.9 Graphical representation of the network node for the corresponding activity

Step 2. Define the earliest start time for any activity using the following rule, "the ES for an activity i is equal to the maximum of the EC_i of the activities immediately preceding activity i"

$$ES_i = \underset{j \in P(i)}{Max} \{EC_j\}$$

Where P (i) = {Set of all the immediate predecessors of activity i}

Step 3. Compute the earliest completion time of activity i. The EC_i is equal to the sum of the activity earliest start time and the estimated activity duration t_i. .

$$EC_i = ES_i + t_i$$

Step 4. Define the estimated project duration. It is equal to the earliest completion time of the very last node, noted n, in the project network.

$$EC_{project} = EC_n$$

b) Backward pass

The backward pass starts by assigning a desired completion time to the very last activity, represented by the node *n* in the project network. Then, the backward pass computations are used to determine the latest allowable starting time (LS_i) and latest allowable completion time (LC_i). The desired completion time for the last activity, which

is also the project desired completion time, may be specified corresponding to a scheduled date defined by the project sponsor or for other reasons. If no scheduled date for the project end is decided, then it is set equal to the earliest completion time of the terminal node *n* in the project network. This is called the zero-slack convention.

$$LC_{project} = EC_n$$

Under this convention, the floats of the activities along the critical path (s) are equal to zero while the floats along the other paths are positive. The concept of float will be discussed later in this chapter. The backward pass calculations are applied using the following steps, going from the project network end node to the initial node:

Step 1 Define the latest allowable completion time for the project terminal activity *n.* it equal to a desired deadline, T_p, if specified or else equal to the earliest completion time of the terminal node n.

$$LC_n = EC_n \text{ or } T_p$$

Step 2 Determine the latest completion time of activity i. The LC_i is equivalent to the smallest of the latest allowable start times of the activities immediately following activity i.

$$LC_i = \underset{j \in S(i)}{\text{Min}} \{LS_j\}$$

Where S (i) = {Set of all the immediate successors of activity i}

Step 3 Compute the latest allowable start time of activity i. The activity i' LS_i is equal to its latest completion time minus its estimated duration t_i

$$LS_i = LC_i - t_i$$

3.3 Slack times/Floats – Critical Path

In the previous paragraph, the concept of float for an activity i was introduced. This float, also referred to as the slack time of activity i, is very important in the CPM procedure, it helps determine the critical activities in a project and thus the critical path. Depending on their interpretation and computations, four basic types of floats can be defined:

- Total Slack (TS). The total slack for activity *i* is the length of time an activity may be deferred from its earliest starting time without negatively affecting, the latest completion time of the project. The TS_i, of an activity i, is equal to the difference between the activity's latest completion and earliest completion times. It is also equivalent to the difference between the activity's latest start and earliest start times.

$$TS_i = LC_i - EC_i = LS_i - ES_i$$

The total slack is the parameter that is used to define the critical activities in a project. They are identified as the activities having a minimum total slack (TS_i).

- Free Slack (FS). The free slack is defined as the interval of time by which an activity may be postponed from its earliest starting time without affecting the earliest start time of any of its immediate successors. It is the difference between the minimum earliest starting time of the activity's immediate successors and the earliest completion time of the activity.

$$FS_i = \underset{j \in S(i)}{\text{Min}} \{ES_j\} - EC_i$$

- Interfering Slack (IS*)*. The interfering slack is the duration by which an activity interferes with its successor when its total slack is totally used. The IS correspond to the difference between the total slack and the free slack.

$$IS_i = TS_i - FS_i$$

- Independent Float (IF). The independent float is the total float that an activity will always have regardless of its predecessors' completion time and successors' starting time. The IF can also be define as the time by which an activity can be delayed without affecting the earliest start of any of its successors, given the fact that the activity's predecessors were concluded at their latest allowable completion time. Mathematically, the IF is defined as follow

$$IF = Max \{0, (ES_j - LC_i - t)\}$$

Where ES_j is the earliest start time of the preceding activities, LC_i is the latest completion time of the succeeding activities and t is the duration of the activity whose independent float is being calculated.

The above-defined floats, mainly the total slack, help determine the critical path in a project network. The critical path is defined as the longest path in the project network diagram and therefore determines the minimum time required to finish the project. A project can have different critical paths but the duration is always the same.

Using the total slack concept, the critical path is the path of activities with the least total slack in the network. If the zero-slack convention is applied, the total slack is equal to zero; therefore the critical path can be defined as the path of activities with a total slack equal to zero.

Activities in the critical path are qualified as critical activities due to the fact that any delays in the completion of these activities cause a postponement of the project completion date.

CHAPTER 4

THE CRITICAL RESOURCE DIAGRAMMING

The Critical Resource Diagramming (CRD) is a simple extension to the CPM technique developed for resource management purposes. Contrary to the CPM method, which focuses on the activities during the scheduling phase, the CRD model concentrates on the resources to schedule the project. It uses nodes to represent each resource unit. Thus, a resource unit may appear more than once in a CRD network, specifying all different tasks for which a particular unit is assigned to.

In the diagram, a node represents a resource unit and has its own duration originated from the resource owner (activity). A node also has links to other nodes based on the technical relationship between activities. The finish-to-start relationship is the only type of precedence relationship used in the CRD program. The backward and forward computations, performed in a CPM procedure, are also applied in the CRD method.

There are many benefits to the use of CRD. A major benefit is that both senior level and lower level resources can be included in the resource-planning network. Another benefit is that during the determination of the task duration for a specific resource, factors such as experience, productivity, etc.... can be incorporated in the computations. This allows a better estimate of a specific task length.

4.1 CRD Network Development

In a CRD, a node is used to represent each resource unit; arrows indicate the interrelationships between resource units. The arrows are referred to as resource-relationship (R-R) arrows. Task durations are included in a CRD to provide further details about resource relationships. A resource unit may appear at more than one location in a CRD network provided there are no time or task conflicts. Such multiple locations indicate the number of different jobs for which the resource is responsible. Figure 4.1 shows an example of a CRD for a small project requiring six different resource types. Each node, RES j, refers to a task responsibility for resource type j.

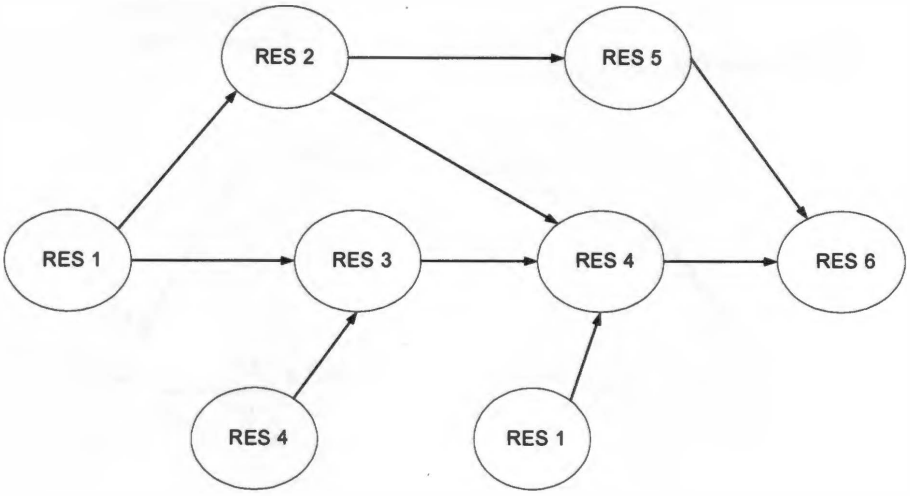

Figure 4.1 Basic Critical Resource Diagram

In Figure 4.1, Resource type 1 (RES 1) and Resource type 4 (RES 4) appears at two different nodes, indicating that each is responsible for two different jobs within the same work scenario. However, appropriate precedence constraints may be attached to the nodes associated with the same resource unit if the resource cannot perform more than one task at the same time.

4.2 CRD Computations & Analysis

The same forward and backward computations used in CPM are applicable to a CRD diagram. However, the interpretation of the critical path may be different since a single resource may appear at multiple nodes. Figure 4.2 presents an illustrative computational analysis of the CRD network. In Figure 4.2, task durations (days) are given below the resource identifications. Earliest and latest times are computed and appended to each resource node in the same manner as in CPM analysis. RES 1, RES 2, RES 5, and RES 6 form the critical resource path. These resources have no slack times with respect to the completion of the given project.

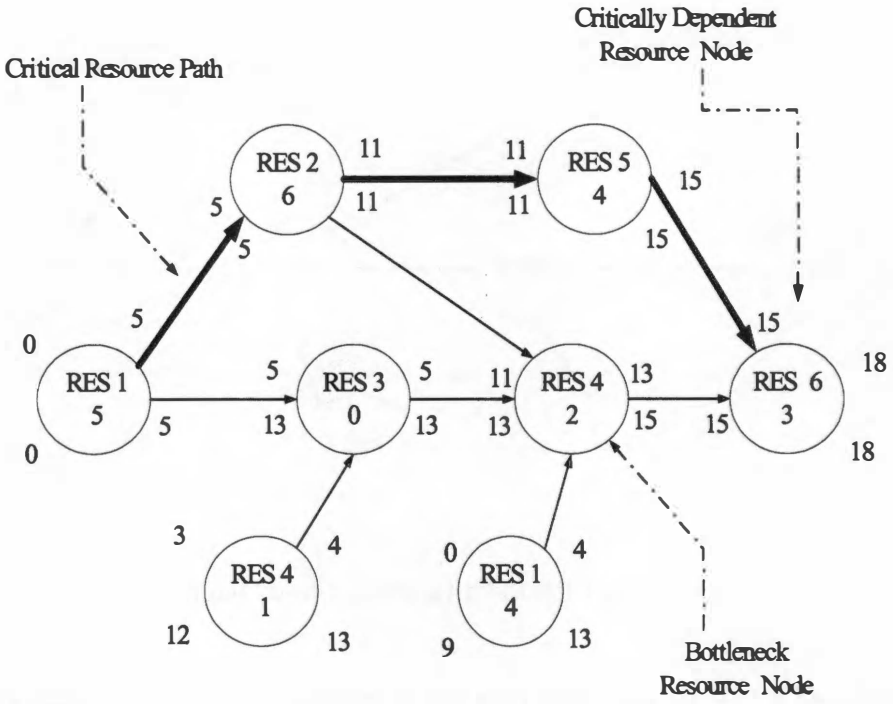

Figure 4.2 CRD Network Analyses

Note that only one of the two tasks of RES 1 is on the critical resource path. Thus, RES 1 has a slack time for performing one job while it has no slack time for performing the other. None of the two tasks of RES 4 is on the critical resource path. For RES 3, the task duration is specified as zero. Despite this favorable task duration, RES 3 may turn out to be a bottleneck resource. RES 3 may be a senior manager whose task is that of signing a work order. But if he or she is not available to sign at the appropriate time then the tasks of several other resources may be adversely affected.

4.3 CRD Node Classifications

A potential *bottleneck* resource node is defined as a node at which two or more arrows merge, commonly called a merge node. In Figure 4.2, RES 3,RES 4, and RES 6 have bottleneck resource nodes. The tasks to which bottleneck resources are assigned should be expedited in order to avoid delaying dependent resources. A *dependent* resource node

is a node whose job depends on the job of immediate preceding nodes. A *critically* dependent resource node is defined as a node on the critical path at which several arrows merge. In Figure 4.2, RES 6 is both a critically dependent resource as well as a bottleneck resource node. As a scheduling heuristic, it is recommended that activities that require bottleneck resources be scheduled as early as possible. A *burst* resource node is defined as a resource node from which two or more arrows emanate. Like bottleneck resource nodes, burst resource nodes should be expedited since their delay will affect several following resource nodes.

CHAPTER 5

METHODOLOGY

5.1 Description

In order to generate a descriptive project schedule given the known resources availability, an approach based on the CPM and the CRD methods was used. The methodology can be illustrated as follows:

a. For each activity, a CRD network is developed and computed to define the activity duration based on its resource's requirements.
b. The calculated durations, for each activity composing the project, are then utilized along with the CPM method to compute the total duration of a project.

The following assumption was also applied during the implementation of the technique:

"For a specific activity, the same resource can only be used once during the execution of the activity".

The latter assumption, which can be very realistic, helped specify another characteristic of the type of projects studied. Cases, where a resource may be used more than once in an activity, can be the subject of further developments of the above approach.

The steps used to apply the presented methodology are as follow:

- Define the project and all of its activities. The project should have only a single start activity and a single finish activity.
- Develop the relationships among the activities. Decide which activities must precede and which must follow others.
- Draw the network connecting all the activities. In this our case, the AON system was used.

- For each activity, define the resource requirements, the estimated time each required resource is involved in the execution of the activity and the precedence relationships between resources
- For each activity, draw its CRD network connecting the different resources and compute the longest time path through the CRD network. This is called the critical resource path. The calculated duration corresponds to the activity duration.
- In the project network, assign each calculated duration to the specific activity.
- Compute the longest time path through the project network. This is called the critical path.
- Use the network to help plan, schedule, monitor and control the project.

5.2 Applications

The project presented in Table 5.1 illustrates a project network analysis using CRD method. The project network is shown in Figure 5.1. The time estimates for each resource to perform a specific task are also presented in Table 5.2 with a time unit in days. The sample project contains six activities and four resource types with each having a quantity of resource *R* equal to 1.

The analysis steps presented above were performed for this sample project. The final critical resource network is described in Figure 5.2. RES 1, RES 2, RES 3 and RES 4 form the critical resource path. They have no slack times with respect to the completion of the given activity. In the example, a dummy resource, DRES, was used in order to respect the network logic. RES 1 is a major resource in the completion of this project. Using the CRD analysis, the estimated completion time of this project equal to thirty-eight days. The estimated activity duration, based on CRD, are summarize in Table 5.3. Figure 5.3 shows the CPM calculations for the entire project.

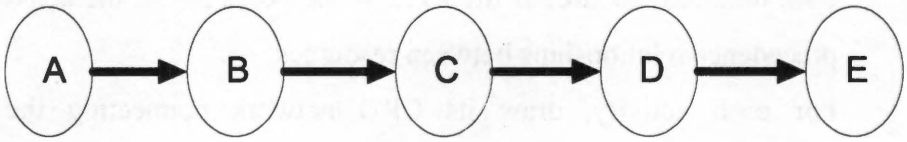

Figure 5.1 Project Network

Table 5.1. Sample project Data

Activity No	Activity	Predecessor	Resources
1	A	-	1
2	B	A	1→ 2, 3
3	C	B	1→ 2
4	D	C	3
5	E	D	1→ 4

Table 5.2 Tasks durations for each resource

Activity No	Res 1	Res 2	Res 3	Res 4
1	4			
2	5	10	5	
3	8	1		
4			4	
5	2			6

Table 5.3 Activities durations under CRD

Activity No	Res 1	Res 2	Res 3	Res 4	Duration
1	4	-	-	-	4
2	5	10	5	-	15
3	8	1	-	-	9
4	-	-	4	-	4
5	2	-	-	6	8

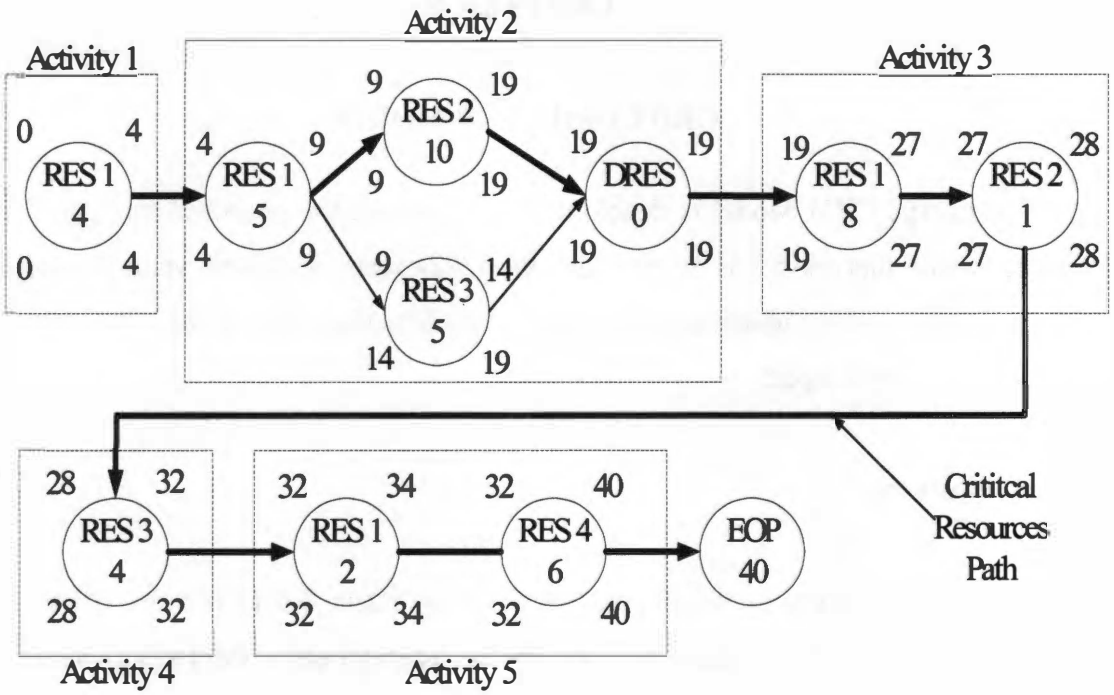

Figure 5.2 CRD network illustrations

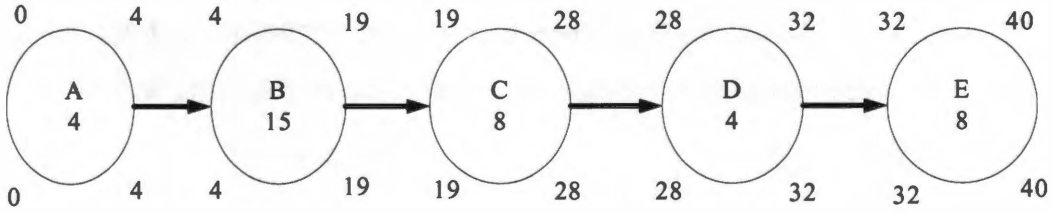

Figure 5.3 CPM network computations

CHAPTER 6

CRD COMPUTER MODEL

A prototype CRD Model is developed to implement the proposed method. The main objective of this model is to evaluate the CRD technique with various project examples to create a better understanding of the studied method limitations. This chapter presents the model developed.

6.1 System Platform

The CRD prototype is developed using Visual Basic 6.0 (VB 6.0). VB is an event-driven programming language and associated development environment, created by Microsoft. It is derived heavily from BASIC. VB enables Rapid Application Development (RAD) of graphical user interface (GUI) applications. The VB language was selected because of its compatibility with the Windows environment. It is also widely used for in-house application program development and prototyping.

The model is composed of two main modules: the GUI and the scheduling module. Figure 6.1 shows the interaction between the different modules during the simulation. A description of each module along with their functionality will be presented later in this chapter.

6.2 Model Overview

The CRD model is implemented on the basis of the event-driven technique. Input from the end-user is necessary in order for the program to run correctly. As stated earlier, the system consists of two modules: the User Interface (UI) and the Scheduling Module (SM).

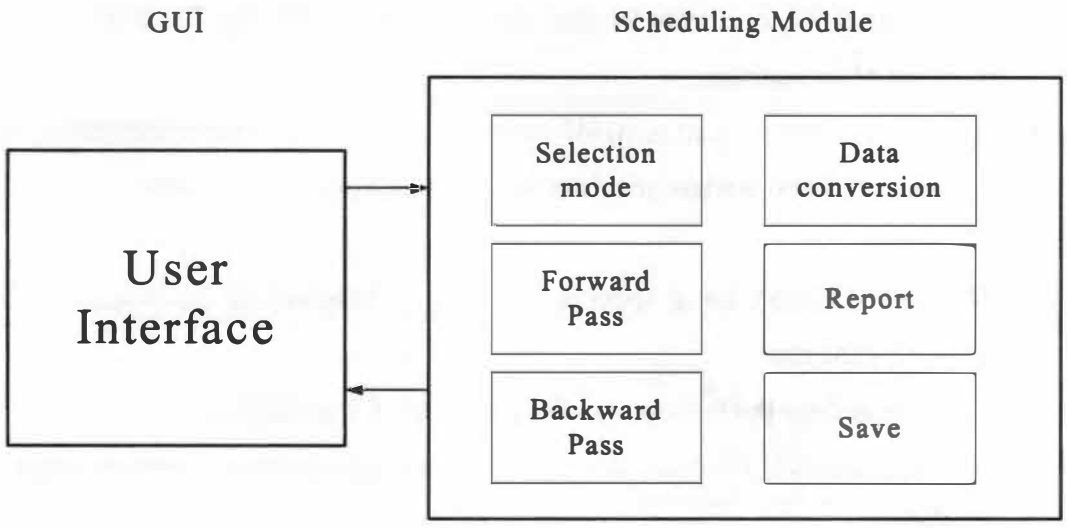

Figure 6.1 Modules interactions in the CRD-model

The User Interface module allows the interaction between the user and the program during the run mode. By means of dialogue boxes, friendly messages and a help option, the user is guided through the program. Data entered are then transferred to the scheduling module

The scheduling module, which is not visible by the user, performs all the necessary computations to develop a schedule. The calculated timetable is presented to the end-user through the UI. Further commands are then available to the user such as "save the output" or "print a report" to keep record of the results.

6.3 Model Description

6.3.1 Assumptions

The assumptions made during the development of the program are as follows:

1. The limited resources are defined in discrete units (e.g. men, machines, trucks).
2. The precedence relationships among activities are well defined and remain constant throughout the project.

3. The resource requirements for each activity are known and fixed throughout the duration of the activity.
4. An activity may not start until all immediate predecessors have been completed.
5. An activity, once started, may not be interrupted (i.e. non preemptive executions of activities).
6. The task duration for a specific resource is assumed to be deterministic and known by the user.
7. The user defines the resource precedence logic for an activity.
8. At the beginning of each activity, the necessary resources are assumed to be available.
9. The finish-to-start relationship is the only type of precedence relationship used in the CRD program.

6.3.2 The Scheduling Module

As shown in the Figure 6.1, the SM consists of six major functions: the mode selection, the forward pass, the backward pass, the save function, the data conversion, and the report function.

1 The mode selection function

Based on the input from the user, this function is used to select the mode under which the program will be running. There are two available approaches in the model:

The mode single activity that allows the user to define a CRD network for a single activity. The inputs required from the user are:

- The resource identification
- The resource task duration, which represents the estimated time necessary to the resource to complete the assignment
- The successor and the predecessors for each resource
- The activity identification

The second approach is the *multiple activities mode*. This approach is selected when the user wants to perform a study on a project. Under this mode, two distinct operations are performed:

- For each activity, a CRD network is developed and computed to define the activity duration based on its resource's requirements.
- The calculated durations, for each activity composing the project, are then used along with the CPM method to compute the total duration of a project.

The inputs required from the user during this phase are:

For the project:

- The activity identification
- The successor and predecessor for each activity

For each activity, the data inputs are similar to those required under the single activity mode.

2 The data conversion function:

The program uses this function to read the data entered by the user.

3 The forward pass

As indicated by its name, this function performs the forward calculation in the CRD model. The results returned by this function are the early start, the early completion time for each resource or activity and the total activity or project duration based on the mode selected.

4 The backward pass

This function executes the backward calculation and returns the latest start, the latest completion, the total float, and the critical level for each resource or activity.

5 The save function

This function creates a number of Microsoft Excel CSV (comma separated values) files of the CRD scheduling outputs. The results are saved using a table format. The saved documents can be accessed again through the CRD model.

6 The report function

This function allows the user to create a report using an html (Hyper Text Markup Language) document. The created document can then be saved or printed depending on the will of the program user.

6.3.3 The User Interface

The CRD model UI was developed using a user-centered design approach. This mandates designing an interface from the view of the user, rather than the view of the system. Based on this method, the UI was created with the following features:

1. Different windows are used for different independent tasks.
2. The interface is consistent in appearance and function: all the window sizes are kept uniform as much as possible and the location on the screen is uniform (center of the screen).
3. Specific clues are displayed throughout the program to help prevent user errors.
4. The error messages are specific and constructive. When entering resources data, the interface prevents the user from entering any alphabets.
5. A help menu will be made available to the user to explain the different functionalities of the model.

Figure 6.2 shows the start page of the CRD model and the three available options. The user can open an already saved file, start a new project, or end the current project. Figure 6.3 and 6.4 also show the CRD user interface in the two modes of operation.

6.3.4 The User Guide

The steps to follow when using the CRD computer model are presented below. Figure 6.5 summarizes the different steps and their interactions.

Step1

Start the program;

Select the running mode project;

If the selected mode is the single activity mode then

Go to step 2

If the selected mode is the multiple activities mode then

Go to step 3

Else

Go to step 4

Step 2

Single activity mode

For each resource, enter resource ID, task duration, resource successors and predecessors;

Forward pass;

Backward pass;

Compute project duration;

Display results;

Show html report;

Then go to step 4.

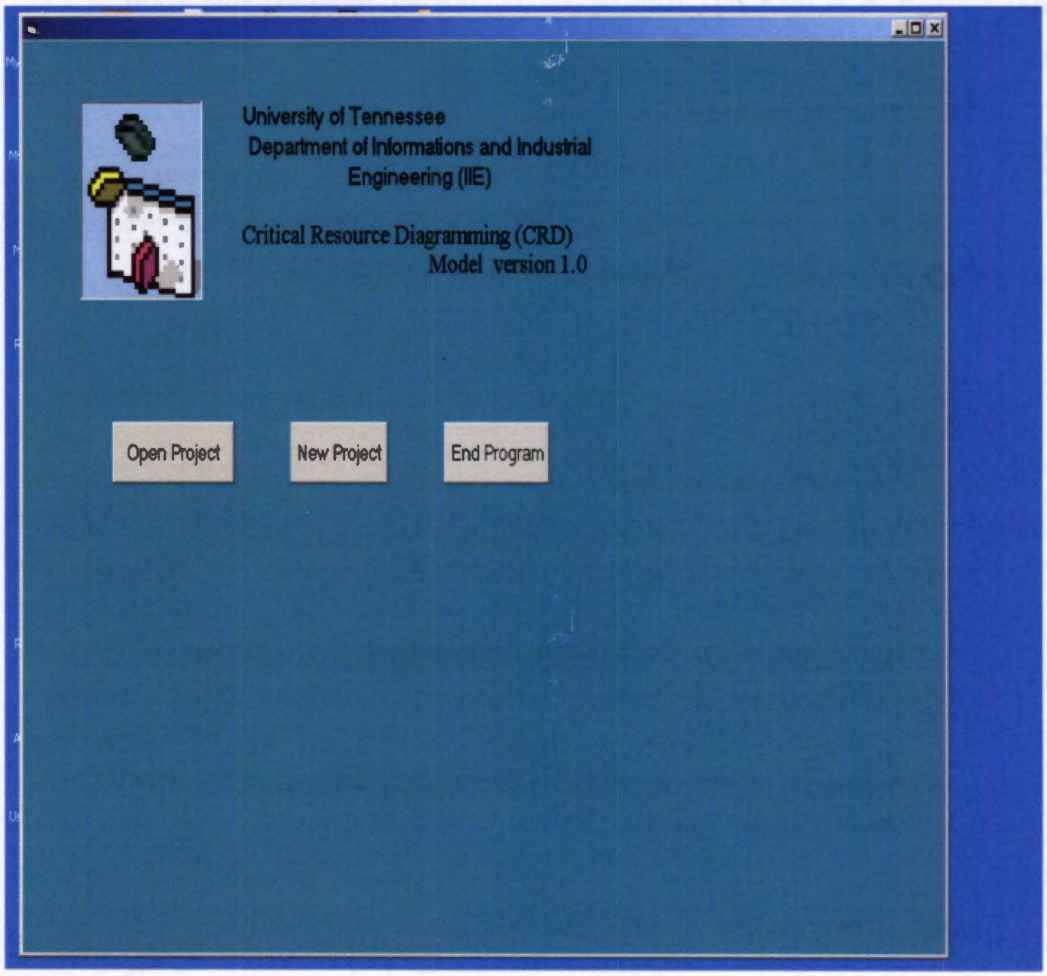

Figure 6.2 The CRD model start window

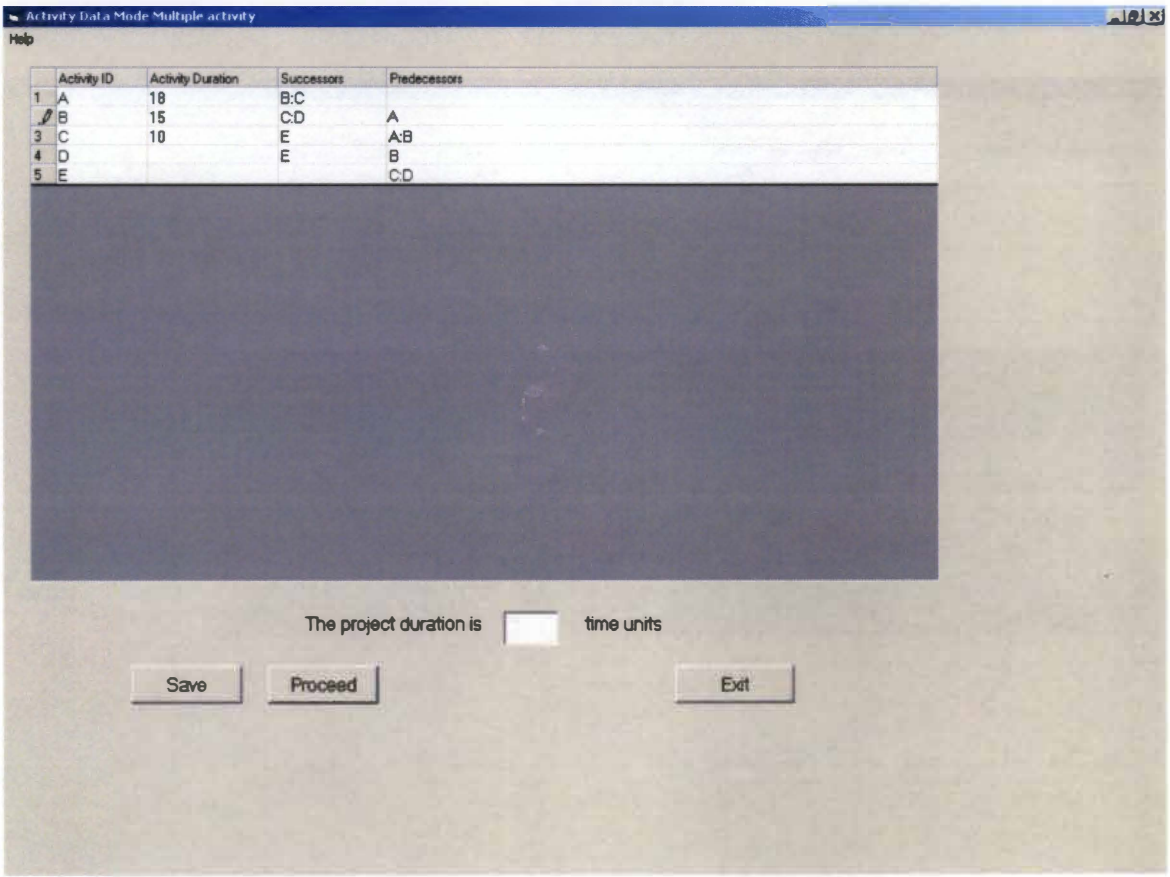

Figure 6.3 The CRD interface under the multiple activities mode

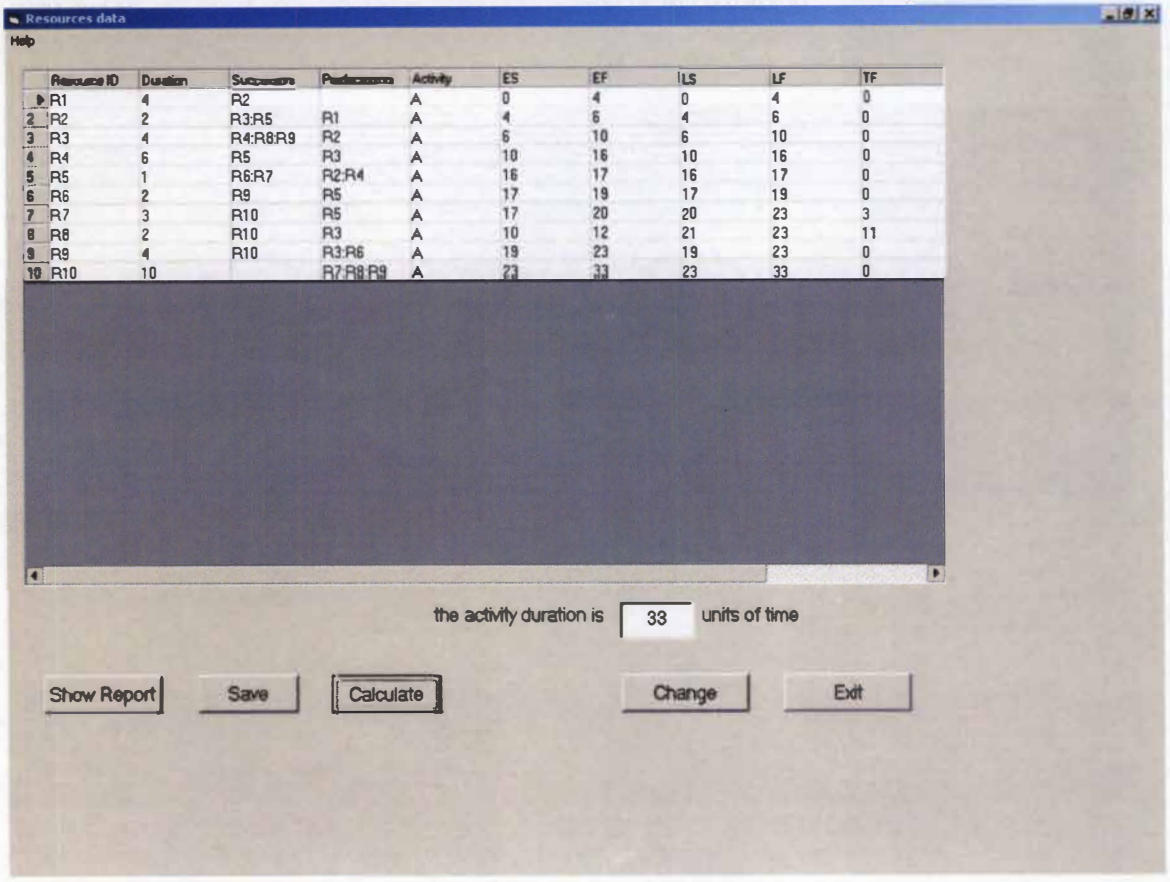

Figure 6.4 The CRD interface under the single activity mode

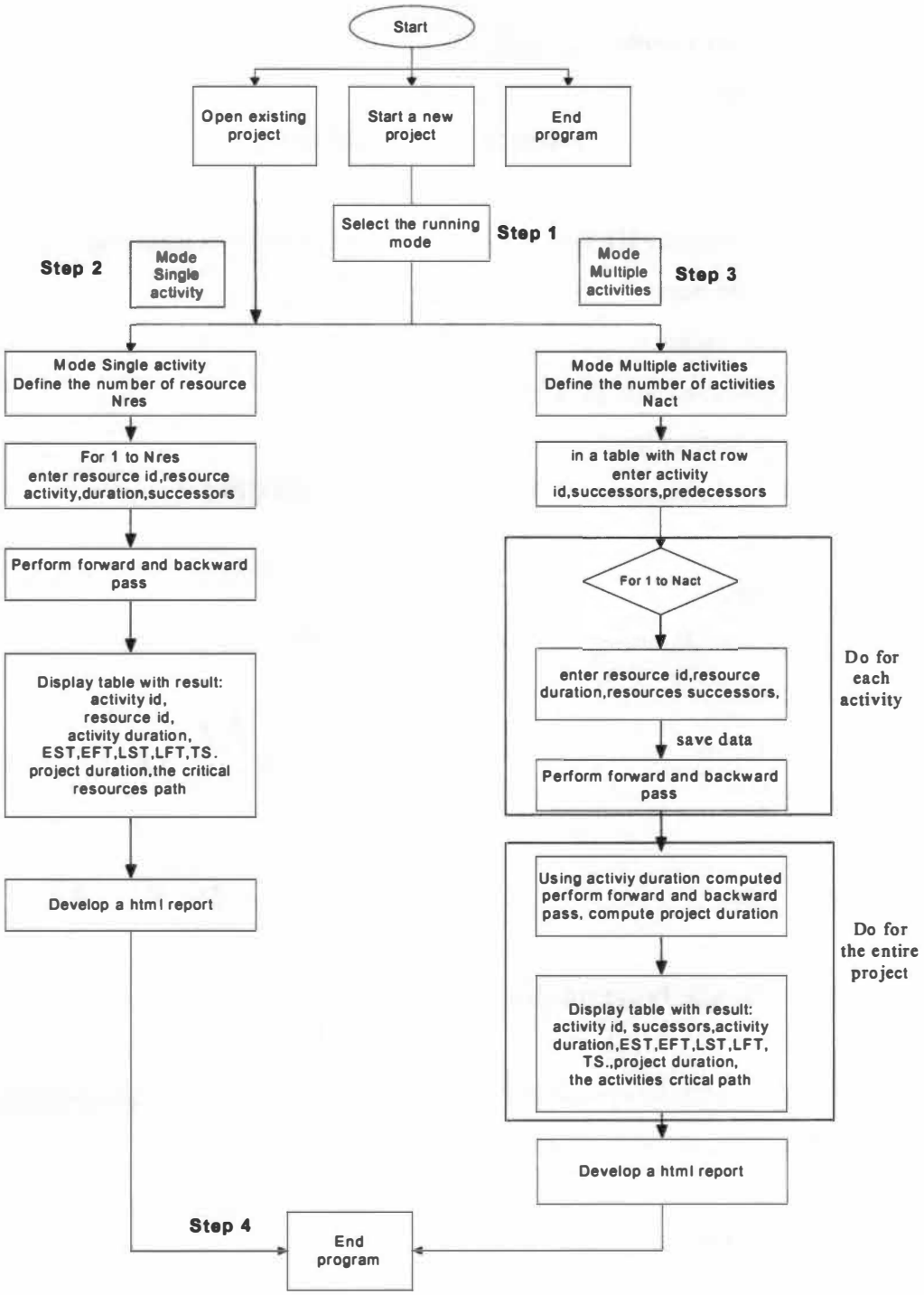

Figure 6.5 CRD program interactions

Step 3

Multiple activities mode

For entire project

Enter activity ID, activity successors and predecessors

For each activity

Enter resource ID, task duration, resource successors and predecessors;

Forward pass;

Backward pass;

Compute activity duration;

Return to the project level

Using activity duration, compute for each activity and CPM algorithm

Forward pass;

Backward pass;

Compute project duration;

Display results;

Show html report;

Then go step 4.

Step 4

End model run.

6.4 CRD Sample Program Run

The project and its data presented in Tables 6.1,6.2 and 6.3 are used to illustrate a project network analysis using the CRD computer model.

6.4.1 Input Data

The example problem mentioned above is used to run the CRD software. The executable version of the program, CRDmethod.EXE, can be run on any personal computer using Microsoft windows 98 or higher as the operating system.

Table 6.1. Project illustration

Activity No	Activity	Predecessor	Resources
1	A	-	1
2	B	A	1→ 2, 3
3	C	B	1→ 2
4	D	C	3
5	E	D	1→ 4

Table 6.2 Tasks durations for each resource in the specific activity

Activity No	Res 1	Res 2	Res 3	Res 4
1	4			
2	5	10	5	
3	8	1		
4			4	
5	2			6

Table 6.3 Calculated activities durations under CRD

Activity No	Res 1	Res 2	Res 3	Res 4	Duration
1	4	-	-	-	4
2	5	10	5	-	15
3	8	1	-	-	9
4	-	-	4	-	4
5	2	-	-	6	8

The following steps should be followed in order to run the example problem:

1. Launch the CRDmethod. The start page should appear with three options: Open project, new project, and end program.
2. Press new project, the mode selection page is then activated.
3. Next select the mode "multiple activities", press submit.
4. A window explaining the functionality of the selected mode will appear. Press Ok to continue.
5. The next activated window will ask you to enter the number of activities in the project, enter "5", then press submit.
6. A grid with the column activity duration shaded will appear and with a number exactly equal to the number of activities. In our case 5 activities, this will correspond to 5 rows.
7. For each activity enter the activity name, the activity predecessors and successors, then press proceed.
8. The program will then perform a CRD analysis for each activity
    a. For each activity, the user will be asked to enter the number of resources, their respective durations, predecessors, and successors.
    b. At the end of each computation, press "next" to go to the next activity.
9. At the conclusion of the activities calculations, the program will display the activity grid, with the activity duration column filled with the corresponding calculated value. Then press, "calculate" to perform the CPM algorithm and determine the project duration.
10. In order to view the study output, the user should press on the "show report" command. A new window with an html report will appear. Use the print or save menu to keep a record of the project.

6.4.2 Output Data

The CRD report of the above example problem is presented in Table 6.4. A notice should be made concerning Table 6.4; the sizes of the tables in the CRD report have been modified in order to fit the page.

The first portion of the report in Table 6.4 shows the CPM analysis result based on the CRD calculation for each activity. The number of activities and the estimated project duration (CRD project duration) is shown. Then, the activity identification (Activity ID), expected CRD duration (Aduration), activity successors (Successors), activity predecessors (Predecessors), earliest start (ES), earliest finish (EF), latest start (LS), latest finish (LF), total float (TF) and the indicator for criticality (CR) for each activity are presented. The output on the example problem gives the following results:

- The CPM project duration of 40 time units
- A critical path with all five activities

The next part of the report presents for each activity the result of the CRD analysis. Similar to the project level, at the activity level the report will display the following information: the activity identification, expected CRD duration and the number of resources used by the activity. Then, the resource identification (Resource ID), expected CRD duration (Rduration), resource successors (Successors), resource predecessors (Predecessors), ES, EF, LS, LF, TF and CR values are shown for the corresponding activity. The results of the example problem show that for activity A, C, D and E each of their resources is critical for their completion and for activity B only Res1 and Res2 are critical for its completion. One should notice also the use of the dummy resource Dres, with duration 0, in the CRD network to create the precedence logic.

Table 6.4 CRD Output for example problem

CRD OUTPUT
MODE Multiple Activities
The CRD project duration = **40 time units**
Number of activities = 5

Activity ID	Aduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
A	4	B		0	4	0	4	0	True
B	15	C	A	4	19	4	19	0	True
C	9	D	B	19	28	19	28	0	True
D	4	E	C	28	32	28	32	0	True
E	8		D	32	40	32	40	0	True

A
the CRD activity duration = **4 time units**
Number of resources = 1

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
Res1	4			0	4	0	4	0	True

B
the CRD activity duration = **15 time units**
Number of resources = 4

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
Res1	5	Res2; Res3		0	5	0	5	0	True
Res2	10	Dres	Res1	5	15	5	15	0	True
Res3	5	Dres	Res1	5	10	10	15	0	False
Dres	0		Res2; Res3	15	15	15	15	0	True

C
the CRD activity duration = **9 time units**
Number of resources = 2

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
Res1	8	Res2		0	8	0	8	0	True
Res2	1		Res1	8	9	8	9	0	True

D
the CRD activity duration = **4 time units**
Number of resources = 1

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
Res3	4			0	4	0	4	0	True

Table 6.4 CRD Output for example problem (continued)

E
the CRD activity duration = **8 time units**
Number of resources = 2

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
Res1	2	Res4		0	2	0	2	0	True
Res4	6		Res1	2	8	2	8	0	True

6.5 Evaluation of the CRD Computer Model

Due to the concept underlying the CRD technique, which required knowledge of the task duration estimate and the dependency relationship for each resource unit involved in an activity or in a project, it was difficult to find suitable problems for a perfect evaluation of the model. Therefore most of the test problems used to assess the model were randomly created. Another option was to use problems encountered in the literature with the activity structure and resource requirements already defined and to modify them to fit the input requirements of the CRD computer model. Such modifications include the incorporation of resource dependency relations and the assignment of task duration for each resource. A wide range of project networks was tested. A presentation of the test problem structure is given in appendix A. The complete CRD outputs for the presented test problems can be found in appendix B and C. All the computations of the CRD software were conducted on the same computer under Windows 2000 environment.

This model evaluation focused first on the ability of the model to produce a schedule, display project and activity duration and indicate to the user the critical resources for each activity in his project. Based on the above-cited criteria, the results of the assessment were satisfying because all the different cases treated by the model generated a report similar to those presented in appendix B and C.

The second focal point was the aptitude of the model to give the exact computational result. In this part of the evaluation, the objective was to certify that the functions created, to calculate the ES, EF, LS, LF, TF and the project duration values, were correctly programmed. The method used was to randomly select in the literature project samples, with their calculated values, and to recalculate the samples using the CRD software. The results were also satisfying. All the computed values using the software were equal to the original results given in the literature. Table 6.5 shows for different project scenarios, the number of activities, the exact project duration and the calculated project duration.

Following the computational evaluation of the CRD model, a benchmarking approach was used to contrast the prototype with existing project management software, such as Microsoft Project 2002. This comparison was motivated by the interest of the developer in finding areas in which the model could be improved. The domains, identified so far, in which the CRD prototype can be enhanced, are as follows.

- Graphical illustrations (i.e. Gantt chart, network diagram) The actual version do not display any type of graph to the user.

- Output reports, for example options in the program that give the resource utilization per activity and on the overall project can be included.

- Costs analysis. A costs report developed along with the resource scheduling output will greatly increase the attractiveness of the developed model.

Table 6.5 Calculated values comparison.

Project	Number of activities	Exact project duration	Calculated project duration
1	4	14	14
2	8	46	46
3	10	30	30
4	11	25	25
5	15	75	75
6	18	93	93
7	20	70	70
8	22	61	61
9	25	55	55
10	30	95	95
11	40	123	123
12	50	200	200

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

This research has illustrated the use of the Critical Resource Diagramming method. The advantages of CRD include simplified resource tracking and control, better job distribution and better information to avoid resource conflicts.

The developed model used both CPM and CRD methods to provide, the project manager, with a set of information (i.e. level of criticality, total floats etc.) on each resource unit under supervision. First a development with the CRD method is executed, to determine the activity duration under the different resource constraints, and then based on the calculated activity duration; the traditional CPM method is applied to generate a project schedule. Subsequently a report is provided to the manager for further analysis of the project. Looking at the output of the CRD model develop, which provides the administrator information on each of the resources under his supervision, one can conclude that the prototype respects the ideas behind the CRD concepts.

The presented tool can also be used to perform some quick "what if " analysis to support or avoid the use of a specific resource unit to perform a critical activity.

The results, from the evaluation stage, were encouraging in the sense that the program solved most of the cases studied, comforting the developer from a technical standpoint.

In summary, the study has made a contribution to resource-constrained project scheduling by producing a simple descriptive scheduling methodology. The developed technique allows the project manager to incorporate the resources constraints in the estimation of an activity duration using the CRD methodology then using the CPM method to generate a descriptive project schedule. However, to make the developed methodology even more powerful in its applications, further research is needed.

7.2 Recommendations for Future Research

The CRD model is based on a number of underlying assumptions. Some of the assumptions are that the task duration for a specific resource unit is deterministic and it is known, the resource is available at the beginning of an activity, the non preemptive executions of activities, there is no uncertainty built in the method. However, these assumptions may not be valid in real world situations. Hence, an attempt should be made to incorporate the following in future research:

- Method to account for uncertainty should be combined with the CRD model
- Preemptions of activities should be allowed
- The user interface should be redesigned for better performances
- An optimization rule should be included in the model.

REFERENCES

REFERENCES

Alvarez-Valdes, R. and J.M. Tamarit " Heuristic algorithms for resource-constrained project scheduling: A review and empirical analysis" in R. Sowinski, J Weglarz (Eds), Advances in Project Scheduling, Elsevier, Amsterdam, 1989, 114-134

Baar, T., P. Brucker and S. Knust, "Tabu search algorithm for the resource-constrained project scheduling problem" in Voss, S., S., Martello, I. Osman and C. Roucairol (eds), Meta-Heuristics: Advances and Trends in local Search Paradigms for Optimization, Klewer, Dordrecht, 1998,1-18

Badiru, Adedeji B., "Quantitative models for Project Planning and Control" Quorum Books, 1993.

Badiru, Adedeji B., "A Project Scheduling Heuristic under Stochastic Time and Resource Constraints (STARC)," doctoral dissertation, University of Central Florida, Orlando, Florida, 1984.

Badiru, Adedeji B., "Critical Resource Diagram: A New Tool for Resource Management," *Industrial Engineering,* Vol. 24, No. 10, 1992, 58-65.

Badiru, Adedeji B., "Activity-Resource Assignments Using Critical Resource Diagramming," *Project Management Journal,* Vol. 24, No. 3,1993,15-21.

Badiru, Adedeji B., and S.P. Pulat, *Comprehensive Project Management: Integrating Optimization Models, Management Principles and Computers,* New Jersey: Prentice Hall, 1995.

Balas, E., "Project Scheduling with Resource Constraints," *Applications of Mathematical Techniques,* Carnegie-Mellon University, Pittsburgh, 1970.

Bedworth, D.D., *Industrial Systems: Planning, Analysis, and Control,* New York: Ronald Press, 1973.

Bedworth, D.D., and J.E. Bailey, *Integrated Production Control Systems: Management, Analysis, and Design,* New York, John Wiley, 1982.

Bell, C.E. and K. Park, "Solving Resource-Constrained Project Scheduling Problems by A* Search, *Naval Research Logistics Quarterly,* Vol. 37, No. 1, 1990, 61-84.

Boctor, F.F " Some Efficient Multi-heuristic Procedures for Resource-Constrained Project Scheduling" *European Journal of operational Research* Vol 49,1990, 3-13

Boctor F. F "A New and Efficient Heuristic for Scheduling Projects with Resource Restrictions and Multiple Executions Modes" *European Journal of operational Research* Vol 90,1996, 349-361

Bottcher, J, A. Drexl, R. Kolish, and F. Salewski "Project Scheduling under Partially Renewable Resource Constraints" *Management Science* Vol 45, No 4,1999

Bouleimen, K and H. Lecocq "A New Efficient Simulated Algorithm for the Resource-Constrained Project Scheduling Problem." Technical Report, Service de Robotique et Automatisation, Universite de Liege, 1998

Brucker, P., A. Drexl, R. Mohring, K. Neumann and E. Pesch "Resource-Constrained Project Scheduling: notation, classification, models, and methods." (1999), European Journal of operational Research Vol 112,1999, 349-361

Cheng, R. and M. Gen "An Evolution Program for the Resource-Constrained Project Scheduling Problem" *international Journal Computer Manufacturing integrated,* Vol 11, No 3, 1998, 274-287.

Cho, J. h. and Y.D. Kim "A Simulated Annealing Algorithm for Resource Constrained Project Scheduling Problem" Journal of the Operational Research Society, Vol 48, 1997,736-744.

Christofides, N.R., Alvarez-Valdes, and J.M Tamarit, "Project Scheduling with Resource Constraints: A Branch and Bound Approach," *European Journal of Operations Research,* Vol. 29, 1987, 262-273.

Cohen, I., A.Mandelbaum and A.Shtub,"Multi-project Scheduling and Control: A Process-Based Comparative Study of the Critical Chain Methodology and Some Alternatives", The Project Management Institute June, Vol 35,No2, 2004,39-50.

Conway, R.W., W.L. Maxwell and L.W Miller " Theory of Scheduling" Addison-Wesley publishing Company, 1967.

Davis, E.W., "Project Scheduling under Resource Constraints: Historical Review and Categorization of Procedures," *AIIE Transactions*, Vol. 5, No. 4,1973, 297-313.

Davis, E. W., and J. H. Patterson, "A Comparison of Heuristic and Optimal Solutions in Resource Constrained Project Scheduling," *Management Science,* Vol. 21, No. 8, 1975, 944-955.

Davis E.W. and G.E. Heidorn, "Optimal Project Scheduling under Multiple Resource Constraints," *Management Science,* Vol. 17, No. 12,1971,803-816.

Demeulemeester, E. and W. Herroelen " A Branch and Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem"*Management Science,* Vol 38, issue 12, 1992,1803-1818

Demeulemeester, E and W. Herroelen " New benchmark Results for the Resource Constrained Project Scheduling Problem" *Management Science*, Vol 43, 1997, 1485-1492

Demeulemeester, E., and E.S. Herroelen, "Project Scheduling A Research Handbook" Kluwer Academic Publishers, 2002.

Drezet, L.E., and C. Tacquard " Multi-Constrained Project Scheduling " Laboratoire d'Informatique, Ecole Polytechnique de l'Universite de Tours, 64 avenue Jean Portalis, 37200 tours 2003

East, W "Infrastructure Management Resource Allocation using Genetic Algorithm" U.S . Army,Corps of Engineers, Construction Engineering Research Laboratories, 2902 Newmark Drive, Champaign, IL 61826-9005, 2001

Goldratt, E. M.,"Critical Chain", The north River Press Publishing Corporation, Great Barrington, MA, 1997.

Elmaghraby, Salah E " Activity Networks: Project Planning and Control by Networks Models" John Wiley and Sons, New York, 1977,156-158.

Elaysed, E.A., "Algorithms for Project Scheduling with Resource Constraints," *International Journal of Production Research,* Vol. 20, No.1, 1982, 95-103.

Glover, F. "Tabu Search Part I" ORSA *Journal on Computing*, 1, 1989,190-206.

Gordon, J. and A. Tulip "Resource Scheduling" *International Journal of Project Management,* Vol. 15, No. 6, 1997,359-370.

Gorenstein S. " An Algorithm for Project Scheduling with Resource Constraints". Operations Research, Volume 20, No 4, July-August 1972, pp 835-850.

Hartmann Sonke, " Project Scheduling under Limited Resources Models, Methods and Applications", Springer 1999

Hegazy, T., "Optimization of Resource Allocation and Leveling using Genetics Algorithm." Journal of Construction Engineering and Management, Vol 125, issue 3, 1999,167-175.

Herroelen, W., E. Demeulemeester, and B.D Reyck," A classification scheme for project scheduling" in Project Scheduling; Recent Models, Algorithms and Applications edited by Jan Weglarz, 1999.

Herroelen, W., R. Leus, and E. Demeulemeester,"Critical Chain Project Scheduling: Do Not Oversimplify", *Project Management Institute,* Vol 33, No 4, 2002,48-60

Holland J.H "Adaptation in a Natural and Artificial Systems" University of Michigan Press Ann Arbor, 1975.

Johnson, T.J.R " An Algorithm for the Resource Constrained Project Scheduling Problem" Doctoral thesis, Massachusetts Institute of Technology, Cambridge, Ma USA, 1967.

Kaplan, L.A.,"Resource-Constrained Project Scheduling via Preemption of Jobs", Unpublished PhD Thesis, University of Michigan, 1988.

Kerzner, Harold, *"Project Management: A Systems Approach to Planning, Scheduling, and Controlling"* Ohio: John Wiley & Sons, Inc., 2003.

Khamooshi H. " Dynamic Priority-Dynamic Programming Scheduling Method (DP)^2 SM: a dynamic approach to resource constraint project" *International Journal of Project Management,* Vol 17, No 6, 1999, 386-391.

Kirkpatrick, S., C. D. Gelatt et al " Optimization by Simulated Annealing" Science 220 1983,671 – 680.

Klein Robert, "Modeling of Resource-Constrained Projects" Kluwer Academic Publishers 2000.

Kohlmorgen, U., H. Schmeck and K. Haase "Experience with Fine-Grained Parallel Genetic Algorithm" Annals of Operations Research, 90,1999,203-219.

Kolisch R., " Efficient Priority Rules for the Resource-Constrained Project Scheduling", *European Journal of Operations Research,* Vol. 14, 1996,179-192.

Kolisch, R. and A. Drexl "Adaptative Search for Solving Hard Project Scheduling Problems" Naval Research Logistics Vol 43,1996, 23-40.

Kolisch, R and S Hartmann "Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis" Print from J. Weglarz (editor): *Project Scheduling: Recent models, algorithms and applications,* Kluwer, Amsterdam, the Netherlands, 1999,147-178.

Lawler, E.L. and D.E. Wood, " Branch and Bounds Methods: A Survey" *Operations Research* volume 14, issue 4, 1966,699 – 719.

Lawrence, S. " Resource Constrained Project scheduling – A Computational Comparison of Heuristic Scheduling Techniques " Technical Report, Graduate School of Industrial Administration, Carnegie Melon University, Pittsburgh, 1985.

Leach, L. P. "Critical Chain Project Management improves project performance" *Project Management journal,* 1999, 39-51.

Lee, J.K and Y.D., Kim, "Search Heuristics for Resource-Constrained Project Scheduling". Journal of the Operations Research Society, 47, 678-689,1996.

Leon V.J and R. Balakrishan " Strength and Adaptability of Problem-Space based Neighborhoods for Resource-Constrained Scheduling" OR Spectrum, Vol 17, 1995, 173-182.

Mingozzi, A., V. Maniezzo, S. Ricciardelli, and L. Bianco " An Exact Algorithm for the Resource-Constrained Project Scheduling Problem based on a New Mathematical Formulation" Management Science, Vol 44, No 5,1998.

Moder, J.J. and C.R. Phillips, *Project Management with CPM and PERT,* New York: Van Nostrand Reinhold Company, 1970.

Naphade, K.S, S.D. Wu and R.H. Storer, " Problem Space search Algorithms for Resource-Constrained Scheduling Problems" Annals of Operations Research, 70, 1997, 307-326.

Neumann, K. and M. Morlock "Operations Research ", Carl Hanser, Munchen, 1993.

Palmer, C. " An Integrated Approach to Manufacturing Planning". University of Huddersfield, 1994.

Patterson J.H, R. Sowinski, F.B Talbot, and J. Weglarz "An algorithm for a general class of precedence and resource constraints scheduling problems" in: R.Sowinski, J Weglarz (Eds), Advances in Project Scheduling, Elsevier, Amsterdam, 1989, 3-28

Patterson, J.H. and G. Roth, "Scheduling a Project under Multiple Resource Constraints: A Zero-One Programming Approach," *AIIE Transactions,* Vol. 8, No. 3, 1976, 449-456.

Pinson, E., C. Prins, and F.Rullier, " Using Tabu Search for Solving the Resource-Constrained Project Scheduling Problem", in *Proceedings of the Fourth International Workshop on Project Planning and Scheduling.* Leuven, 12-15 July 1994,102-106

Pritsker, A.A., L. J. Waters and P. M. Wolfe, "Multiproject Scheduling with Limited Resources: A 0-1 approach" *Management Science*, volume 16, 1969, 93-108

Raz, T., R. Barnes and D.Dvir" A Critical Look at Critical Chain" *Project Management Institute*, Vol 34,No 4, 2003, 24-32.

Selle, T and J. Zimmermann " A Bidirectional Heuristic for Maximizing the Net Present Value of Large-Scale Projects Subject to Limited Resources" Wiley Periodicals, Inc *Naval Research Logistics* Vol 50, 2003, 130-148.

Senouci A.B. and H.Adeli, "Resource Scheduling using Neural Dynamics Model of Adeli and Park ", *Journal of Construction Engineering and* Management, Vol 127, No 1, 2001.

Shixin, L and W. Mengguang " An object-oriented methodology for solving the RCPSPs with heuristics and Metaheuristics" *Production Planning and Control*, Vol 11, No 5, 2000, 434-442.

Slowinski R, Blazewicz, J., W. Cellary, J. Weglarz,. "Scheduling under Resource Constraints--Deterministic Models". *Annals of Operations Research* 7,1986,1-35.

Sprecher, A " Scheduling Resource-Constrained Projects Competitively at Modest Memory Requirements" *Management Science* Vol 46, No 5, 2000, 710-723

Sprecher, A., R. Kolish and A. Drexl," Semi Active, Active and Non-Delay Schedules for the Resource-Constrained Project Scheduling Problem" European Journal of Operational Research, 80, 1995, 94-102.

Sprecher, A., and A.Drexl "Manuskripte aus den Instituten fur Betriebswirtschaftslehre der Universitat Kiel", Kiel, Universitat Kiel, 1996.

Stinson, J.P, E.W. Davis, and B.M. Khumawala, "Mulstiple Resource-Constrained Scheduling using Branch and Bound" *AIIE transaction* 10, 1978, 252-259.

Talbot, F.B., "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case," *Management Science,* Vol. 28, No. 10,1982, 1197-1210.

Talbot, F.B. and J.H. Patterson, "An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems," *Management Science,* Vol. 24, No. 11, 1978,1163-1174.

Thesen, A "Heuristic Scheduling of Activities Under Resource and Precedence Restrictions" *Management Science* Vol 23, No 4, 1976, 412-422

Ulusoy, G and L. Özdamar "Heuristic Performance and Network/Resource Characteristics in resource-constrained project scheduling" Journal of the Operations Research Society, Vol 40, No 12, 1989,1145-1152.

Wall M.B., " A Genetic Algorithm for Resource-Constrained Scheduling" doctoral dissertation, Massachusetts Institute of Technology, Massachusetts 1996

Whitehouse, G.E., and J.R. Brown, "GENRES: An Extension of Brooks Algorithm for Project Scheduling with Resource Constraints," *Computers and Industrial Engineering,* Vol. 3, No. 4, 1979, 261-268.

Wiest, J.D., and F.K. Levy, *A Management Guide to PERT/CPM,* New Jersey: Prentice Hall, Inc., 1969

APPENDICES

APPENDIX A

DATA FOR TEST PROBLEM

PROBLEM TEST 1 (GENERATED AT RANDOM)

			Task Duration for each resource									
Activity	Successors	Predecessors	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
A	B		2	5	0	6	1	8	2	5	2	3
B	C,D,E,F,G	A	0	6	3	0	1	0	3	6	8	5
C	E,F,G	A,B	1	9	3	3	1	2	8	2	5	1
D	H,I	B,C	2	0	6	8	1	0	9	8	0	2
E	J,I	B,C	6	2	5	5	1	2	5	3	1	0
F	H	B,C	9	3	8	7	1	6	2	0	6	5
G	I,H,J	B,C	0	0	2	2	1	0	0	3	3	0
H	I,J	G,F,D	5	2	6	5	1	2	2	5	6	4
I	J	H,G,E,D	6	3	0	0	1	6	0	4	1	0
J		I	0	2	2	0	1	2	5	0	2	2

Activity	Resource sequence
A	R1,R2 → R4,R5,R6→ R7,R8→ R9, R10
B	R3,R2 → R5,R7→ R8→ R9, R10
C	R1 → R2,R3,R4* → R5,R6,R7→ R8,R9→ R10
D	R1 → R3,R4→ R5→ R7,R8→ R10
E	R1,R3 → R2,R5,R6→ R4,R8→ R9,R7
F	R1,R2,R3 → R4,R5,R6→ R9,R10
G	R3 → R4→ R5,R8→ R9
H	R1,R2 → R4,R5,R6→ R7,R8→ R9, R10,R3
I	R1,R2 → R5,R6→ R8→ R9
J	R2,R3 → R5,R6,R7→ R9,R10

* R2, R3, R4 means that the task start time for each of the resources is the same.

PROBLEM TEST 2 (ADAPTED FROM BADIRU [1984])

			Duration for each resource type (availability					
Activity	Successors	Predecessors	R1	R2	R3	R4	R5	R6
A	B,C,D		2	5	0	6	1	8
B	E,F,G	A	0	6	3	0	1	0
C	E,F,G	A	1	9	3	3	1	2
D	E,F,G	A	2	0	6	8	1	0
E	H,I,J	B,C,D	6	2	5	5	1	2
F	H,I,J	B,C,D	9	3	8	0	1	6
G	H,I,J	B,C,D	0	0	2	2	1	0
H	K,L	E,F,G	5	2	6	5	1	2
I	K,L	E,F,G	6	3	0	2	1	0
J	K,L	E,F,G	0	2	2	0	1	2
K	L	H,I,J	2	0	0	2	1	0
L		H,I,J,K	0	3	5	0	1	3

Activity	Resource sequence
A	R1 → R2,R4→ R5→ R6
B	R2→ R3,R5
C	R1 → R2→ R3→ R4→ R5→ R6
D	R1 → R3,R4→ R5
E	R1 → R2,R3→ R4→ R5,R6
F	R1 → R2,R3→ R6,R5
G	R3→ R4→ R5
H	R1 ,R2→ R3→ R4,R5→ R6
I	R1 ,R2→ R4→ R5
J	R2→ R3→ R5→ R6
K	R1 → R3→ R4
L	R2→ R3→ R5,R6

APPENDIX B

CRD OUTPUT FOR PROBLEM TEST 1

CRD OUTPUT
MODE Multiple Activities
the CRD project duration = 137 time units
Number of activities = **10**

Activity ID	Aduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
A	21	B;C		0	21	0	21	0	True
B	23	C;D;E;F;G	A	21	44	21	44	0	True
C	24	D;E;F;G	A;B	44	68	44	68	0	True
D	22	H;I	B;C	68	90	68	90	0	True
E	18	J;I	B;C	68	86	93	111	25	False
F	22	H	B;C	68	90	68	90	0	True
G	10	I;H;J	B;C	68	78	80	90	12	False
H	21	I;J	G;F;D	90	111	90	111	0	True
I	17	J	H;G;E;D	111	128	111	128	0	True
J	9		I	128	137	128	137	0	True

A
the CRD activity duration = **21 time units**
Number of resources = **11**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES*	0	R1;R2		0	0	0	0	0	True
R1	2	R4;R5;R6	DRES	0	2	3	5	3	False
R2	5	R4;R5;R6	DRES	0	5	0	5	0	True
R4	6	R7;R8	R1;R2	5	11	7	13	2	False
R5	1	R7;R8	R1;R2	5	6	12	13	7	False
R6	8	R7;R8	R1;R2	5	13	5	13	0	True
R7	2	R9;R10	R4;R5;R6	13	15	16	18	3	False
R8	5	R9;R10	R4;R5;R6	13	18	13	18	0	True
R9	2	DRES2	R7;R8	18	20	19	21	1	False
R10	3	DRES2	R7;R8	18	21	18	21	0	True
DRES2*	0		R9;R10	21	21	21	21	0	True

* Dummy resources use to maintain the network logic.

B

the CRD activity duration = **23 time units**

Number of resources = **9**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R3;R2		0	0	0	0	0	True
R3	3	R5;R7	DRES	0	3	3	6	3	False
R2	6	R5;R7	DRES	0	6	0	6	0	True
R5	1	R8	R3;R2	6	7	8	9	2	False
R7	3	R8	R3;R2	6	9	6	9	0	True
R8	6	R9;R10	R5;R7	9	15	9	15	0	True
R9	8	DRES1	R8	15	23	15	23	0	True
R10	5	DRES1	R8	15	20	18	23	3	False
DRES1	0		R9;R10	23	23	23	23	0	True

C

the CRD activity duration = **24 time units**

Number of resources = **10**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	1	R2;R3;R4		0	1	0	1	0	True
R2	9	R5;R6;R7	R1	1	10	1	10	0	True
R3	3	R5;R6;R7	R1	1	4	7	10	6	False
R4	3	R5;R6;R7	R1	1	4	7	10	6	False
R5	1	R8;R9	R2;R3;R4	10	11	17	18	7	False
R6	2	R8;R9	R2;R3;R4	10	12	16	18	6	False
R7	8	R8;R9	R2;R3;R4	10	18	10	18	0	True
R8	2	R10	R5;R6;R7	18	20	21	23	3	False
R9	5	R10	R5;R6;R7	18	23	18	23	0	True
R10	1		R8;R9	23	24	23	24	0	True

D

the CRD activity duration = **22 time units**

Number of resources = **7**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	2	R3;R4		0	2	0	2	0	True
R3	6	R5	R1	2	8	4	10	2	False
R4	8	R5	R1	2	10	2	10	0	True
R5	1	R7;R8	R3;R4	10	11	10	11	0	True
R7	9	R10	R5	11	20	11	20	0	True
R8	8	R10	R5	11	19	12	20	1	False
R10	2		R7;R8	20	22	20	22	0	True

E

the CRD activity duration = **18 time units**

Number of resources = **11**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R3		0	0	0	0	0	True
R1	6	R2;R5;R6	DRES	0	6	0	6	0	True
R3	5	R2;R5;R6	DRES	0	5	1	6	1	False
R2	2	R4;R8	R1;R3	6	8	6	8	0	True
R5	1	R4;R8	R1;R3	6	7	7	8	1	False
R6	2	R4;R8	R1;R3	6	8	6	8	0	True
R4	5	R9;R7	R2;R5;R6	8	13	8	13	0	True
R8	3	R9;R7	R2;R5;R6	8	11	10	13	2	False
R9	1	DRES1	R4;R8	13	14	17	18	4	False
R7	5	DRES1	R8;R4	13	18	13	18	0	True
DRES1	0		R7;R9	18	18	18	18	0	True

F

the CRD activity duration = **22 time units**

Number of resources = **10**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R2;R3		0	0	0	0	0	True
R1	9	R4;R5;R6	DRES	0	9	0	9	0	True
R2	3	R4;R5;R6	DRES	0	3	6	9	6	False
R3	8	R4;R5;R6	DRES	0	8	1	9	1	False
R4	7	R9;R10	R1;R2;R3	9	16	9	16	0	True
R5	1	R9;R10	R1;R2;R3	9	10	15	16	6	False
R6	6	R9;R10	R1;R2;R3	9	15	10	16	1	False
R9	6	DRES1	R4;R5;R6	16	22	16	22	0	True
R10	5	DRES1	R4;R5;R6	16	21	17	22	1	False
DRES1	0		R9;R10	22	22	22	22	0	True

G

the CRD activity duration = **10 time units**

Number of resources = **5**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R3	2	R4		0	2	0	2	0	True
R4	2	R5;R8	R3	2	4	2	4	0	True
R5	1	R9	R4	4	5	6	7	2	False
R8	3	R9	R4	4	7	4	7	0	True
R9	3		R5;R8	7	10	7	10	0	True

H

the CRD activity duration = **21 time units**

Number of resources = **12**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R2		0	0	0	0	0	True
R1	5	R4;R5;R6	DRES	0	5	0	5	0	True
R2	2	R4;R5;R6	DRES	0	2	3	5	3	False
R4	5	R7;R8	R1;R2	5	10	5	10	0	True
R5	1	R7;R8	R1;R2	5	6	9	10	4	False
R6	2	R7;R8	R1;R2	5	7	8	10	3	False
R7	2	R9;R10;R3	R4;R5;R6	10	12	13	15	3	False
R8	5	R9;R10;R3	R4;R5;R6	10	15	10	15	0	True
R9	6	DRES2	R7;R8	15	21	15	21	0	True
R10	4	DRES2	R7;R8	15	19	17	21	2	False
R3	6	DRES2	R7;R8	15	21	15	21	0	True
DRES2	0		R9;R10;R3	21	21	21	21	0	True

I

the CRD activity duration = **17 time units**

Number of resources = **7**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R2		0	0	0	0	0	True
R1	6	R5;R6	DRES	0	6	0	6	0	True
R2	3	R5;R6	DRES	0	3	3	6	3	False
R5	1	R8	R1;R2	6	7	11	12	4	False
R6	6	R8	R1;R2	6	12	6	12	0	True
R8	4	R9	R5;R6	12	16	12	16	0	True
R9	1		R8	16	17	16	17	0	True

J

the CRD activity duration = **9 time units**

Number of resources = **9**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R2;R3		0	0	0	0	0	True
R2	2	R5;R6;R7	DRES	0	2	0	2	0	True
R3	2	R5;R6;R7	DRES	0	2	0	2	0	True
R5	1	R9;R10	R2;R3	2	3	6	7	4	False
R6	2	R9;R10	R2;R3	2	4	5	7	3	False
R7	5	R9;R10	R2;R3	2	7	2	7	0	True
R9	2	DRES1	R5;R6;R7	7	9	7	9	0	True
R10	2	DRES1	R5;R6;R7	7	9	7	9	0	True
DRES1	0		R9;R10	9	9	9	9	0	True

APPENDIX C

CRD OUTPUT FOR PROBLEM TEST 2

CRD OUTPUT
MODE Multiple Activities
the CRD project duration = 93 time units
Number of activities = **12**

Activity ID	Aduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
A	17	B;C;D		0	17	0	17	0	True
B	9	E;F;G	A	17	26	27	36	10	False
C	19	E;F;G	A	17	36	17	36	0	True
D	18	E;F;G	A	17	35	18	36	1	False
E	11	H;I;J	B;C;D	36	47	48	59	12	False
F	23	H;I;J	B;C;D	36	59	36	59	0	True
G	5	H;I;J	B;C;D	36	41	54	59	18	False
H	18	K;L	E;F;G	59	77	59	77	0	True
I	9	K;L	E;F;G	59	68	68	77	9	False
J	7	K;L	E;F;G	59	66	70	77	11	False
K	5	L	H;I;J	77	82	77	82	0	True
L	11		H;I;J;K	82	93	82	93	0	True

A
the CRD activity duration = **17 time units**
Number of resources = **5**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	2	R2;R4		0	2	0	2	0	True
R2	5	R5	R1	2	7	3	8	1	False
R4	6	R5	R1	2	8	2	8	0	True
R5	1	R6	R2;R4	8	9	8	9	0	True
R6	8		R5	9	17	9	17	0	True

B

the CRD activity duration = **9 time units**

Number of resources = **4**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R2	6	R3;R5		0	6	0	6	0	True
R3	3	DRES	R2	6	9	6	9	0	True
R5	1	DRES	R2	6	7	8	9	2	False
DRES	0		R3;R5	9	9	9	9	0	True

C

the CRD activity duration = **19 time units**

Number of resources = **6**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	1	R2		0	1	0	1	0	True
R2	9	R3	R1	1	10	1	10	0	True
R3	3	R4	R2	10	13	10	13	0	True
R4	3	R5	R3	13	16	13	16	0	True
R5	1	R6	R4	16	17	16	17	0	True
R6	2		R5	17	19	17	19	0	True

D

the CRD activity duration = **18 time units**

Number of resources = **7**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	6	R2;R3		0	6	0	6	0	True
R2	2	R4	R1	6	8	9	11	3	False
R3	5	R4	R1	6	11	6	11	0	True
R4	5	R5;R6	R2;R3	11	16	11	16	0	True
R5	1	DRES	R4	16	17	17	18	1	False
R6	2	DRES	R4	16	18	16	18	0	True
DRES	0		R5;R6	18	18	18	18	0	True

E

the CRD activity duration = **11 time units**

Number of resources = **4**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	2	R3;R4		0	2	0	2	0	True
R3	6	R5	R1	2	8	4	10	2	False
R4	8	R5	R1	2	10	2	10	0	True
R5	1		R3;R4	10	11	10	11	0	True

F

the CRD activity duration = **23 time units**

Number of resources = **6**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	9	R2;R3		0	9	0	9	0	True
R2	3	R6;R5	R1	9	12	14	17	5	False
R3	8	R6;R5	R1	9	17	9	17	0	True
R6	6	DRES	R2;R3	17	23	17	23	0	True
R5	1	DRES	R2;R3	17	18	22	23	4	False
DRES	0		R6;R5	23	23	23	23	0	True

G

the CRD activity duration = **5 time units**

Number of resources = **3**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R3	2	R4		0	2	0	2	0	True
R4	2	R5	R3	2	4	2	4	0	True
R5	1		R4	4	5	4	5	0	True

H

the CRD activity duration = **18 time units**

Number of resources = **7**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R2		0	0	0	0	0	True
R1	5	R3	DRES	0	5	0	5	0	True
R2	2	R3	DRES	0	2	3	5	3	False
R3	6	R4;R5	R2;R1	5	11	5	11	0	True
R4	5	R6	R3	11	16	11	16	0	True
R5	1	R6	R3	11	12	15	16	4	False
R6	2		R5;R4	16	18	16	18	0	True

I

the CRD activity duration = **9 time units**

Number of resources = **5**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
DRES	0	R1;R2		0	0	0	0	0	True
R1	6	R4	DRES	0	6	0	6	0	True
R2	3	R4	DRES	0	3	3	6	3	False
R4	2	R5	R1;R2	6	8	6	8	0	True
R5	1		R4	8	9	8	9	0	True

J

the CRD activity duration = **7 time units**

Number of resources = **4**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R2	2	R3		0	2	0	2	0	True
R3	2	R5	R2	2	4	2	4	0	True
R5	1	R6	R3	4	5	4	5	0	True
R6	2		R5	5	7	5	7	0	True

K

the CRD activity duration = **5 time units**

Number of resources = **3**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R1	2	R3		0	2	0	2	0	True
R3	2	R4	R1	2	4	2	4	0	True
R4	1		R3	4	5	4	5	0	True

L

the CRD activity duration = **11 time units**

Number of resources = **5**

Resource ID	Rduration	Successors	Predecessor	ES	EF	LS	LF	TF	CR
R2	3	R3		0	3	0	3	0	True
R3	5	R5;R6	R2	3	8	3	8	0	True
R5	1	DRES	R3	8	9	10	11	2	False
R6	3	DRES	R3	8	11	8	11	0	True
DRES	0		R5;R6	11	11	11	11	0	True

Vita

Vincent Gilles Delgado was born in Dakar, Senegal. He grew up in Dakar and graduated from L'Institution Notre Dame High School in 1996. He then attended the University Cheikh Anta Diop-Ecole Superieure Polytechnique in Dakar, and obtained a Diploma d'Ingenieur de Conception in Eletromechanical Engineering in July of 2001. He started worked for ASIT as a project engineer. In August 2002, he entered the master's program at the University of Tennessee, Knoxville, and worked as a graduate assistant for Dr. Adedeji Badiru. He received his Master's degree in Industrial Engineering in December 2004.

5901 7111 9