T. C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

PROJECT SCHEDULING USING BEE COLONY OPTIMIZATION

Ayşe Ayçim SELAM

THESIS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN INDUSTRIAL ENGINEERING

SUPERVISOR
Prof. Dr. Ercan ÖZTEMEL

İSTANBUL 2011

T. C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

PROJECT SCHEDULING USING BEE COLONY OPTIMIZATION

Ayşe Ayçim SELAM
(141200920030484)

THESIS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN INDUSTRIAL ENGINEERING

SUPERVISOR
Prof. Dr. Ercan ÖZTEMEL

İSTANBUL 2011

ACKNOWLEDGEMENT

It is a pleasure to thank those who made this thesis possible, with their inspiration, support, understanding, guidance and dedication; to only some of whom it is possible to give a particular mention here.

First and foremost, I would like to thank my supervisor, Prof. Dr. Ercan ÖZTEMEL. I owe him my deepest gratitude not only for his endless encouragement, guidance, support and dedication from the initial to the final level, but also for his good advice, invaluable friendship and moral support. He was the one enabling me to develop an understanding of the subject, advising with patience at every step of the thesis, and passing me to the other side whenever I run into a stone wall.

I owe sincere and earnest thankfulness to Prof. Dr. Harun TAŞKIN and Asst. Prof. Dr. Serol BULKAN as my thesis assessment committee for sparing their valuable time to inspect my thesis, and sharing their constructive comments for the improvement of the subject.

I was delighted to interact with Ebubekir KOÇ, without whose interviews it would be impossible to understand some main concepts of the Bees Algorithm.

I am also indebted to Recep KÜCEK for assisting me with his invaluable advice and supervision in coding, answering me patiently and giving critical proposals for the computer code of the model.

This thesis would not been possible without my colleagues, who made Marmara University, Industrial Engineering Department a convivial place to work. It is my pleasure to thank my teammates, the Research Assistants of the Industrial Engineering Department, for their friendship, support, motivation and inspiration throughout my thesis. Their personal therapies were soothing and I knew that I would find one or the other around to comfort me whenever I needed.

I cannot find words to express my gratitude to my very dear and special friend M. Övül Arıoğlu Salmona, who was with me at every second of my doctorate. She was the one who has always been a constant source of inspiration, motivation, and encouragement with her academic and emotional support. I thank her gratefully for riding with me on this road of doctorate and giving me strength for moving on.

My parents and family deserve special mention for their support, motivation and prayers. Special thanks to my mother and father, Fatma and İbrahim Atalay for raising me with their caring, gentle love and for taking good care of my daughter when I was away; my aunt Yurdagül Atalay for encouraging me in my academic career; my uncle Turan Atalay for his valuable advices on academic life; and my brothers Ahmet Kubilay Atalay and Sercan Kutluay Atalay who held my hand whenever I needed.

I owe my heartfelt thanks to my husband Mesut Selam for his dedication, love, patience, faithful support and persistent confidence in me. Without his love and support it would be impossible for me to go through the roughest times. I am grateful to him for being an understanding life partner and encouraging support in my academic career.

I am indebted to my beloved daughter Melike Sedef Selam who joined us during my dissertation; I owe her all the moments which I had to spend on my thesis instead of her. Her presence has always been a roborant support to me. I once more thank to my dearest daughter for her patience, love and understanding.

Finally, I would like to thank everybody who was important to the successful realization of this thesis, as well as expressing my apology to those that I could not mention here personally one by one.

This thesis is dedicated to my grandmother Hatice Atalay who was the greatest reason that I acquired a profession as a woman and my grandfather İsmail Atalay, who always supported me throughout my education and thanks to the greatest moments I have lived with the pleasure of having adoring grandparents.

TABLE OF CONTENTS

PAGE NO

ACKNOWLEDGEMENT i

TABLE OF CONTENTS iii

ÖZET v

ABSTRACT vi

CLAIM FOR ORIGINALITY vii

SYMBOLS ix

ABBREVIATIONS x

LIST OF FIGURES xi

LIST OF TABLES xii

CHAPTER I. INTRODUCTION 1

CHAPTER II. PROJECT SCHEDULING 4

II.1. THE NEED FOR A PROJECT SCHEDULING METHOD 7

II.2. CHALLENGES OF PROJECT MANAGEMENT AND SCHEDULING 10

CHAPTER III. BEE COLONY OPTIMIZATION 13

III.1. PARAMETERS OF BCO 14

III.2. INTRODUCTION OF BCO TO LITERATURE 15

III.3. APPLICATIONS OF BCO IN LITERATURE 17

CHAPTER IV. PROJECT SCHEDULING WITH BEE COLONY OPTIMIZATION ALGORITHM (A PROPOSED MODEL) 23

IV.1. PROJECT SCHEDULING WITH BEES 23

IV.2. DEFINITIONS 23

IV.3. FORAGER ALLOCATION MECHANISM FOR PROJECT SCHEDULING PROBLEM 24

IV.4. MULTI – MODE PROJECT SCHEDULING PROBLEMS 25

IV.5. A PROJECT SCHEDULING PROBLEM 26

IV.6. UTILIZATION OF BA FOR PROJECT SCHEDULING PROBLEM 29

CHAPTER V. REAL-TIME PROJECT TRACKING 32

V.1. REAL-TIME SYSTEMS 32

V.2. REAL-TIME SCHEDULING 33

V.3. REAL-TIME TRACKING OF THE SAMPLE PROJECT 36

CHAPTER VI. BEE-BASED REAL-TIME SCHEDULING MODEL 40

CHAPTER VII. IMPLEMENTATION AND EXPERIMENTS 47

VII.1. FINDING THE SHORTEST COMPLETION TIME WITH BA . 50

VII.2. REAL-TIME PROJECT TRACKING WITH BA 53

CHAPTER VIII. CONCLUSIONS AND FUTUREWORKS 59

REFERENCES 61

APPENDIX 68

CURRICULUM VITAE 97

ÖZET

ARI KOLONİSİ OPTİMİZASYONU KULLANARAK PROJE ÇİZELGELEME

Proje planlama ve çizelgeleme kısıtlı kaynak mevcudiyeti söz konusu olduğunda oldukça karmaşık hale gelmektedir. Maliyet açısından verimli proje çizelgeleri oluşturabilmek, bu açıdan sistematik yaklaşımlar gerektirmektedir. Bu tezde karmaşık optimizasyon problemlerini çözmek için kullanılan Arı Kolonisi Optimizasyonu, bu amaçla kullanılan bir yaklaşım olarak sunulmaktadır. Arı kolonisi Optimizasyonunun proje çizelgeleme problemlerini çözebilecek bir yöntem olup olmadığı incelenmiştir. Bu amaçla Arı Kolonisi tabanlı optimizasyon modeli oluşturulmuştur. Önerilen model örnek bir problem üzerinde çalıştırılarak kavramsal bir doğrulama gerçekleştirilmiştir.

Oluşturulan modelin önemli bir özelliği de gerçek-zamanlı proje izlemeyi mümkün kılmasıdır. Söz konusu proje birçok alternatif içinden en uygun çizelge ile çizelgelense de, gerçek zamanda izlenmekte ve projenin çizelgelendiği gibi devam etmesine engel olan herhangi bir durum oluştuğunda, sistem projeyi yeni koşullar altında yeniden çizelgeleyebilmektedir. Bu çalışmanın kullanım alanı, proje çizelgelemenin yapıldığı ve proje bazlı çalışan, yazılım geliştirmeden inşaat sektörüne kadar uzanan geniş bir alandır. Bu yüzden önerilen model üretim, bilgisayar teknolojileri, inşaat, makine dizaynı, çevre ve şehir projeleri gibi birçok alana uygulanabilecek niteliktedir.

Ağustos, 2011 **Ayşe Ayçim Selam**

ABSTRACT

PROJECT SCHEDULING USING BEE COLONY OPTIMIZATION

Project planning and scheduling is an extremely complex problem due to the limited amount of resources available. Generating cost effective project schedules therefore requires systematic methodologies. In this study Bee Colony Optimization which is a new tool to solve complex optimization problems is utilized as a good alternative for this purpose. The aim of this research was therefore set to investigate the possibility of utilizing Bee Colony Optimization for project scheduling problems. Bee Colony based optimization model is generated and the proposed model is used on a case study for the proof of concept.

Another important characteristic of the proposed model is that it can perform real-time project tracking where after scheduling the project among many alternatives, it will be tracked in real-time and if any kind of interruption occurs a new schedule reflecting the current situation is generated. In other words a reschedule for the project is generated to adapt to the new environment, new defined conditions. The scope of this study is wide from software development to construction as project scheduling is used in such areas where mainly projects are carried out. The proposed method can therefore be applied to many projects some of which are in production, computer technologies, civil engineering, machine design, environmental, and urban project areas.

August, 2011 **Ayşe Ayçim Selam**

CLAIM FOR ORIGINALITY

PROJECT SCHEDULING USING BEE COLONY OPTIMIZATION

One of the main problems with the approaches in "project scheduling" literature is the difficulty of solving combinatorial optimization problems. Another problem with project scheduling problems is that they require heuristic approaches for solving problems of practical size. There is also a need for optimal approaches for validating, comparing and benchmarking solutions given by heuristics. Uncertainty of activities, resources and durations which can change in accordance with real life implementation and utilization of activities and resources still require more efficient and effective methodologies. Solving and revising project scheduling problems, especially in a resource constrained environment, in a countable time is nearly impossible.

Researcher created several algorithms and heuristics for planning and scheduling of the projects. The most well-known of these in literature are ant colony optimization, genetic algorithms, swarm intelligence, classical metaheuristics, and nonstandard metaheuristics etc. Besides, the software industry also developed and marketed some user-friendly computer programs such as Microsoft Project, Primavera, or ZOHO Projects for this purpose. However, these tools seem to be far off to the changing requirements of the project management domain, as well as changing conditions of a particular project. These types of software tools have the ability of visualizing project schedules nicely; however, they do not seem to possess the ability to decide among the alternative solutions. The requirement of high number inputs for software program is another challenge; most of the time schedulers do not possess this type of information. It is obtained from other departments whom change their minds about the durations frequently; rapid adaptation to these changes is a tough situation.

The aim of this study is therefore to investigate a new approach which combines local search with neighborhood search so called Bee Colony Optimization. The solution of project scheduling problem using Bee Colony Optimization Algorithm is investigated. This biologically inspired approach is currently being employed mainly to solve continuous optimization problems (Pham et al., 2006a), scheduling problems (Chong et al., 2006; Pham et al, 2007d; Pham and Koç, 2008; Wong et al., 2008c), to utilize training efficiency in neural networks (Pham et al., 2006b; Pham et al., 2006c; Pham et al., 2006d; Pham et al., 2006e; Pham and Sholedolu, 2008; Öztürk and Karaboğa, 2008; Pham and Haj Darwish, 2010) to optimize mechanical and electronic component design (Pham et al., 2007c; Pham et al., 2008a; Pham et al., 2008b; Karaboğa and Akay, 2010), and to generate solution to traveling salesman problem (Wong et al., 2008a; Wong et al., 2008b). In addition to these, even if some scheduling problems are researched in literature, multi-mode project scheduling is new for utilizing this optimization technique.

A project scheduling model using a new method called Bee Colony Optimization is defined in this research; however, an additional approach to track the projects in real-time is needed. Although real-time scheduling systems can be found in literature, a real-time tracking for bee-based, multi-mode, resource-constrained project scheduling problem will be a new domain in project management. In order to check if the project is being realized using the pre-designed schedule, an appropriate tracking system for this purpose is defined. As projects are realized in real-time and sudden changes along the project life may affect the project flow critically, a scheduler also needs a real-time tracking system giving quick response to these changes. The proposed model emphasizes on a real-time tracking capability. With the combination of both project scheduling and real-time tracking bee-based real-time project scheduling model is developed which is original in its nature.

August, 2011 **Prof. Dr. Ercan Öztemel** **Ayşe Ayçim Selam**

SYMBOLS

a : Critical path factor

A_j : Resource availability factor

e : Number of best sites out of m selected sites

k : Number of activities to be completed

l : number of resources available at time t

m : Number of selected sites

n : Number of scout bees

nep : Number of bees recruited for the best **e** sites

ngh : Initial size of the patches

nsp : Number of bees recruited for the other **m-e** sites

n(SS) : Element number of solution space

p_{ij} : processing time of ith job on j^{th} resource

r_i : Number of resources for activity i

R_i : i^{th} resource

S : Schedule

S' : Reschedule

SS : Solution space of alternative completion ways for the project

TC : Total completion time

ABBREVIATIONS

ABC : Artificial Bee Colony
ACO : Ant Colony Optimization
ACS : Ant Colony System
AS : Ant System
BA : Bees Algorithm
BCO : Bee Colony Optimization
BCOi : Bee Colony Optimization Improved
BS : Bee System
CPM : Critical Path Method
FBS : Fuzzy Bee System
GA : Genetic Algorithms
GERT : Graphical Evaluation and Review Technique
HS : Harmony Search
MCF : Manufacturing Cell Formation
MLP : Multi-Layered Perceptrons
PERT : Project Evaluation and Review Technique
PID : Proportional-Integral-Derivative
PSIEA : Particle Swarm Inspired Evolutionary Algorithm
PSO : Particle Swarm Optimization
PSP : Project Scheduling Problem
RBF : Radial Basis Function
SA : Simulated Annealing
SI : Swarm Intelligence
SVM : Support Vector Machine
TS : Tabu Search
TSP : Traveling Salesman Problem

LIST OF FIGURES

PAGE NO

Figure IV.1. Project Network of the Sample Problem 27

Figure IV. 2. Network of Best Resource Utilization for the Sample Project .. 31

Figure V.1. Project schedule network after generating a reschedule 39

Figure VI.1. Bee-Based Real-Time Scheduling Model 40

Figure VI.2. Gantt Chart of the Sample Project 44

Figure VI.3. Gantt Chart for the Sample Project in Crisis 45

Figure VI.4. Reschedule of the Sample Project with Resource Crisis 46

LIST OF TABLES

PAGE NO

Table III.1. Pseudo code of the basic bee algorithm 14

Table III.2 Theory Building Researches on BCO 16

Table III.3. Scheduling Applications with BCO 18

Table III.4. Neural Network Applications with BCO 19

Table III.5. Robotics Applications with BCO 20

Table III.6. TSP Applications with BCO 20

Table III.7. Other applications with BCO 21

Table IV.1. Forager Allocation Mechanism for PSP 25

Table IV.2. Activity Information for Sample PSP 27

Table IV.3. Job durations for resources 28

Table IV.4. A part of the project pool 29

Table IV.5. The "**nep**" Solutions for the 1st Iteration 30

Table IV.6. The "**nsp**" Solutions for the 1st Iteration 30

Table IV.7. Best resource utilization for the sample problem 31

Table V.1. 1st Resource utilization after generating a reschedule 38

Table V.2. 2nd Resource utilization after generating a reschedule 38

Table VI.1. An Example of Source File 42

Table VII.1. Project completion times of 10-job projects 50

Table VII.2. Project completion times of 20-job projects 51

PAGE NO

Table VII.3. Project completion times of 30-job projects 52

Table VII.4. Project completion times of 50-job projects 52

Table VII.5. Project completion times of 80-job projects 53

Table VII.6. A 10-job project real-time tracking 54

Table VII.7. A 20-job project real-time tracking 55

Table VII.8. A 30-job project real-time tracking 56

Table VII.9. A 50-job project real-time tracking 57

Table VII.10. An 80-job project real-time tracking 58

CHAPTER I

INTRODUCTION

A project can be considered as failed if it is not completed on-time, in-budget, and up to the contracted specifications. The success of a project can be assured through a cost effective project schedule. Project scheduling, on the other hand, is a complex task having a severe effect on project activities. Not being able to generate an effective schedule clearly increases the rate of failure. Taking this into account, it can also be understandable why the number of project failures is much more than that of successes. Generating a timely implementable schedule complying with the specified project plan is therefore one of the most important issues in sustaining cost effective project management. So far as project scheduling is a significant area for project management, new methods investigated can be applied to software development, construction, production, computer technologies, civil engineering, machine design, environmental, and urban design, etc. where projects are carried out.

In order to schedule a project, the project environment must be well-understood. If there is a single way for processing each job of the project, most of the time the problem is to schedule the project under a limited availability of the resources. However, nowadays, more than one way to complete the project activities can be defined, as the technology provides new processes for performing activities. As well as technology, the rich variety of machinery equipment of the companies and the number of subcontractors for outsourcing the project activities adds new ways of performing the project. When the activities can be performed in more than one way, the problem is to select an effective single mode to complete the project on time, in budget and under the contracted specifications. The selection of a single activity mode among alternatives to execute the activities is called the multi-mode project scheduling problem.

There have been several studies on creating algorithms and heuristics for planning and scheduling of the projects. Some of them are Ant Colony Optimization

(ACO), Swarm Intelligence (SI), priority rule-based X-pass methods, classical metaheuristics, nonstandard metaheuristics etc. Besides, there are some user-friendly computer programs such as Microsoft Project, Primavera, or ZOHO Projects which were developed and marketed for this purpose. However, these tools are hardly enough to be applicable for changing requirements of the environment as well as changing resources of a particular project. Furthermore these programs do not seem to possess the ability to decide among the alternative solutions. The main motivation of these software systems is just to visualize the project activities. They also require high number of inputs about the project which will have to be provided by the scheduler who may not always find them available.

Apart from dynamic requirements and high number of inputs, there is another problem that has to be faced. That is the time required to solve the problems, and the applicability, efficiency and accuracy of the methods employed. Since, the problem is highly dynamic, complex and extra resource as well as constraints are inevitable, the project scheduling is considered to be an "np-hard problem" (Lancaster and Özbayrak, 2007). Another problem is the difficulties and hardness of changing the schedules under any change in resources, durations or respective activities. In order to overcome these problems some methodologies are proposed. One of them could be Bee Colony Optimization (BCO).

Note that, a great number of traditional engineering models and algorithms which are used to solve complex project problems are based on control and centralization. Various natural systems (social insect colonies) such as bees indicate that very simple individual organisms can create systems which are able to perform highly complex tasks by dynamically interacting with each other (Teodorovic et. al., 2006).

On the other hand, the BCO is a tool to solve complex optimization problems. The algorithm is inspired by the food foraging behavior of honey bees and could be regarded as belonging to the category of "intelligent" optimization tools (Pham et. al., 2006a). It is a search algorithm capable of locating good solutions as efficiently as possible. This biologically inspired approach is currently being employed to solve continuous optimization problems, to utilize training efficiency in neural networks, to optimize mechanical and electronic component design, to sort out combinatorial optimization problems such as job shop scheduling, and to generate solution to

Traveling Salesman Problem (TSP), etc (Nakrani and Tovey, 2004). The aim of the study is to investigate the capability and applicability of BCO for Project Scheduling Problem (PSP).

Content of the thesis includes first a brief overview of project scheduling. An extensive literature survey of project scheduling and basic problems is provided in this context. The Third Chapter introduces information about the Bee Colony Optimization algorithm, its capabilities and application areas. Feasibility and possible benefits of project scheduling with Bees Algorithm (BA) is elaborated with a proposed scheduling model, with its main problems and implementation in the Fourth Chapter. The Fifth Chapter extends third chapter with a real-time project tracking system. In the Sixth Chapter a bee-based project scheduling model with a real-time tracking capability is introduced. The Seventh Chapter obtains implementation of the model and the experiments conducted. The last Chapter provides the futureworks, discussions and conclusions which includes information about the results.

CHAPTER II

PROJECT SCHEDULING

A project is typically said to be a one-of-a-kind effort undertaken for the purpose of achieving a specific objective. Project is defined by Gray and Larson (2008) as follows:

"*Project is a complex, nonroutine, one-time effort limited by time, budget, resources, and performance specifications designed to meet customer needs.*"

The major characteristics of a project are;

- An established objective,
- A defined life span with a beginning and an end,
- Usually the involvement of several departments and professionals,
- Typically, doing something that has never been done before, and
- Specific time, cost and performance requirements.

Project management is a set of principles, methods and techniques applied for the purpose of keeping the project on-time, under-budget and up to the contracted specification. One of the major functions of the project management is to generate of a project schedule (Knotts et al., 2000). Scheduling is defined as a decision making process with the allocation of scarce resources to tasks over time to optimize one or more objectives (Pinedo, 2002).

The management of large projects requires some tools capable of performing analytical analysis especially for scheduling activities and allocating the related resources. Some tools and techniques developed for this purpose are Critical Path Method (CPM), Project Evaluation and Review Technique (PERT), Graphical Evaluation and Review Technique (GERT). Most widely used and known of these tools are CPM and PERT. PERT was developed by the U.S. Navy and its consultants for the Polaris missile project, while the critical path method was created by Dupon and the Remington Rand Corporation for the management of large chemical plants

(Shumsky, 2003). These tools are used in many project scheduling areas changing widely from construction to software development.

The project scheduling literature largely concentrate on the generation of a precedence and resource feasible schedules optimizing the respective objective(s) (which is most often the project duration) (Herroelen and Leus, 2005). The key activities at the start of a project are planning and scheduling (Knotts et al., 2000). If this phase of the project management is carried out poorly, then the probability for the failure of the project increases.

Traditionally, the PSPs are formulated through the activities, resources, precedence relations, and performance measures. A project consists of a number of activities, also known as jobs, operations, or tasks. Often technological reasons imply that some activities have to be finished before others can start.

The types of resources that may be used during projects are renewable resources, nonrenewable resources, doubly constrained resources and partially renewable resources. Well-known objectives of the project scheduling are makespan minimization, net present value maximization, quality maximization and cost minimization, etc (Pinedo, 2002).

A proper definition of activities provides the information needed for graphical representation of predicted tasks, milestones, dependencies, resource requirements, task duration, and deadlines. The graphical representation of project plan components creates the project schedule. Like the development of each components of a project plan, developing a schedule is an iterative process including the following activities:

- Defining the type of the schedule,
- Defining the precise and measurable milestones,
- Estimating task durations,
- Defining priorities,
- Defining the critical path,
- Documenting assumptions,
- Identifying risks, and
- Reviewing results (California Technology Agency, 1997).

Given above can be referred as a standard project schedule development process. There have been lots of methods used to schedule the projects. Different

algorithms and heuristics are used to solve different kinds of problems in the literature.

The types of problems that is dealt with in scheduling concept are; capital constrained problem, time/cost trade-off resource constrained project scheduling problem, multiple execution modes, scheduling with minimal and maximal time lags, and the payment scheduling, etc.

In capital constrained problem the case is a capital constraint that limits the value of work that may be put in progress at any time (Kolisch and Padman, 2001).

In the classical resource-unconstrained time/cost trade-off problem, there exists for each activity a continuous and linear cost-duration relationship. A natural extension to this problem is the resource-constrained discrete time/cost trade-off problem (Kolisch and Padman, 2001).

In multiple execution modes the multi-mode problem is characterized by the fact that activities can be performed in different discrete modes. Hence, a solution has to determine the timing of activities as in traditional scheduling, the assignment of modes (Kolisch and Padman, 2001). This makes the problem even more complex and finding a solution becomes an NP-hard optimization problem.

All of the models and methods treated allow only the in-corporation of minimal time lags of the finish-to-start type between pairs of activities. When maximal time lags are allowed, the problem becomes much more complex and standard resource constrained project scheduling problem algorithms generally fail to obtain solutions (Kolisch and Padman, 2001).

When organizing projects, a firm is committing its financial resources to that project. Because of this the contractor needs periodic (or at some milestones) payments in order to continue the project activities and increase profitability. However, the client has to possess the objectives of the project before having done all the payments. This problem, which is equally relevant for contractors and clients alike, is called the Payment Scheduling Problem (Kolisch and Padman, 2001).

Many algorithms are used in both resource unconstrained case and the resource constrained case for different types of problems given above and much more others. Different conditions and project environments are investigated using these algorithms.

The recent heuristics used for project scheduling in the literature are priority rule-based X-pass methods, classical metaheuristics, nonstandard metaheuristics and other methods (Kolisch and Hartmann, 2006). Sampling is an example of an x-pass method which is a global-biased random sampling approach. This method employs the serial schedule generation scheme. Well-known classical metaheuristics used in project scheduling are Genetic Algorithms (GA), Tabu Search (TS), Simulated Annealing (SA), and Ant Systems (AS). There are some approaches which can be viewed as metaheuristics although they do not follow one of the classic metaheuristic schemes. These nonstandard metaheuristics are local search-oriented approaches and population-based approaches. There are some other heuristics that can neither be classified as X-pass construction methods nor as metaheuristics. These methods use forward-backward improvement and some further heuristics methods (Kolisch and Hartmann, 2006).

One of the main problems with the approaches in literature is the difficulty of solving combinatorial optimization problems. Another problem with project scheduling problems is that, they require heuristic approaches for solving problems of practical size. There is also a need for optimal approaches for validating, comparing and benchmarking solutions given by heuristics. Another problem with the project scheduling, apart from finding a single solution, is the uncertainty of activities, resources and durations which can change in accordance with according to the revision of the activities and resources. Solving and revising project scheduling problems, especially in a resource constrained environment, in a countable time is nearly impossible.

The aim of this study is to investigate a new approach which combines local search with neighborhood search so called BCO. The solution of project scheduling problem using BCO algorithm is investigated through this research. The discussion and conclusion section highlight the results of the study and the futurework that might be subject to the future studies.

II.1. THE NEED FOR A PROJECT SCHEDULING METHOD

During project realization sudden changes may occur which affects the project conditions seriously. Available tools which are used to schedule the projects and visualize conditions seem to have the lack of adaptability to those sudden changes.

For instance, the project completion date which is predefined in the project plan restricts the project time duration to certain limits. This may restrict the schedule and sudden changes may be difficult to encounter. Besides, the tools and methods to be employed require a high degree of scheduling knowledge and domain experience. Before establishing a project network there is a possibility of setting wrong dependencies, unexpected violation of constraints, and unforeseen inadequate level of details. Over-allocated resources also make it even harder to plan the respective projects. Apart from these, the methods employed by different scheduler may create different schedules even for the same project.

As stated earlier the first phase of the project schedule development is the definition phase including the definition of type of the schedule, precise and measurable milestones, activities, resources that will be used for the given activities. While defining the type of the schedule a detailed investigation must be made for the choice of right type, else new problems may occur while revising the schedule. The establishment of milestones is as important as the "type", because the milestones must be precise and measurable; else problems in project check out times might be inevitable. Lastly the definition of activities is the most important of all because they show which steps that the project will go through. The proper level must be chosen. A very detailed plan for a long term project or a rough plan for short term projects will not clearly show the project steps properly. There is a methodology to identify respective resources. Because, after activity definition the tools used for scheduling doesn't have the ability to choose the resource by itself. Tools used in market require the knowledge of the scheduler and they demand the scheduler to choose the respective resource. However if the resource alternatives are provided to the tool and the tool chooses the idle or the proper resource for the activity, the problem of choosing resources through using scheduler's knowledge would not be needed. For example, if a project manager manages 6 projects and every project has an average of 20 activities; the scheduler has to manage totally 120 activities. Suppose 120 activities he/she has 3 resources in hand, then he/she has to deal with 3^{120} alternatives. This situation gives a numerous alternatives and makes it nearly impossible to find the best solution.

For the example above project activity durations are assumed to be certain. In a situation where the activity durations estimated are not exact and can change during

the project realization, project environment may become a total mess. In most of the time, the scheduler has to revise the plan for every change considering the task durations, resource allocations, and due dates. During a project if resources are adequate but the demand varies widely, evening out resource demand by delaying noncritical activities to lower peak demand and increase resource utilization is called resource leveling (Gray and Larson, 2008). For instance, the Ms Project Software solves the conflicts using the "resource leveling" command. This execution reschedules the project plans but the new due dates can be carried to a date that is highly forward from the previous one. This automatically generated solution actually does not solve the problem of the scheduler. The new schedule must be prepared manually with the new conditions. However, dealing with a high number of inputs manually such as projects, tasks, durations, resources, and due dates is a tough situation to overcome.

Another problem with uncertainty is the risk management concept through project management. Good scheduling makes allowances for risks. The success of the project depends on the schedule which is prepared with foreseeing the risks. So that even the risks occur during project realization, scheduler may have precautions in hand to implement on the new situation. Similarly, documentation of assumptions is also critical for the final success of the project. Without clear documentation of assumptions later changes to the schedule becomes very difficult and risky to follow.

Shortly the main problems in project scheduling are proper definition of schedule type, milestones, activities, and resources; estimation of durations; leveling of resources without the knowledge of the scheduler; identifying risks and documenting assumptions. The needs of a new project scheduling method are quick response to changes caused by estimation and definition problems and the ability to cope with the respective problems of leveling of resources and uncertainty of risks.

To overcome most of the above problems, there is a need for new approaches and more applicable methodologies to uncertain, dynamic, continuously changing and extremely complex environments. BCO is considered to be an alternative source of solution to these kinds of problems in project scheduling. It is designed on the bases of bee behavior as in Chapter III.

II.2. CHALLENGES OF PROJECT MANAGEMENT AND SCHEDULING

Project management involves coordination of various elements such as personnel, materials, procedures, and facilities of a project in order to bring forth a positive result. The project management challenges can be caused by both internal and external factors. Some of these internal factors of project management that are particularly challenging are maintaining the quality that the clients need, qualification of staff, cost, contracting of the work, stretch targets and schedule. Besides these internal factors there are some external factors which project managers cannot change the effects of them but have to deal with. One of these external factors is availability of resources both as labor and materials. However this is an external factor it affects project schedule seriously. Another external factor is the work hours of the vendors which can be different from the parent company and this can cause the schedule delays and respective cost increase. In addition to these there is another factor which the project managers can have exactly no control over is weather. Extreme weather conditions can cause irresistible delays for the project completion. Lastly, another challenge that a project manager has to deal with from time to time is economic crisis around the world. Economic crisis all around the world affects the global economies and the companies directly. While dealing with crisis companies have to hold clients in hand, find new projects to survive during the crisis and finish the projects in order to get the payments under the hard condition of the crisis. (Doucet, 2007)

This research will only deal with scheduling aspect which is one of the internal challenges of project management domain. A list of challenges effecting the project management and scheduling in several ways are (Ford, 2004) :

- *Unrealistic deadlines:* Faulty initial deadlines may cause schedule slippage, high cost due to delays and being discredited in the sight of client.
- *Communication deficit:* Not providing enough or related information and improper communication flow may bring the projects to a deadlock.
- *Scope challenges:* This is the tendency of a project to include more tasks or to implement more systems than originally specified, which

often leads to higher than planned project costs and an extension of the initial implementation date.

- *Resource competition:* Projects usually compete for resources (people, money, time) against other projects and initiatives. This competition brings constraints to resources and makes the scheduling problem in an insolvable situation.
- *Uncertain dependencies:* While determining dependencies schedulers do not take risks into account or the reliability behind the linkages depend on the trusting someone else's experience. In this situation risks may always occur and those experienced people's point of view may always change. Project completion is again brought to a bottleneck because of risks and unreliable experience.
- *Failure to manage risk:* A project plan has included in it some risks simply listed, but no further review happens unless instigated by an event later on.
- *Insufficient team skills:* Team members for many projects are assigned based on their availability but they may not be trained for these jobs. Improper team members may delay the project.
- *Lack of accountability:* The project participants and related team members are not held accountable for their results. Not being responsible for the jobs they have undertaken relaxes people. Because of this people do not care for the jobs they do and make so many mistakes.
- *Customers and end-users are not engaged during the project:* Project teams can get wound up in their own world of internal deliverables, deadlines, and process, and the people on the outside do not get to give added input during the critical phases.
- *Visions and goals not well-defined:* The goals of the project, along with the sub-projects or major tasks involved, are not always clearly defined.

In addition to these challenges Egeland (2007) lists the top 5 challenges he met during his project management experience as the following;

- Lack of project management involvement during the sales process,
- Unrealistic schedule expectations,

- Renegade team members,
- Financial management / budgeting, and
- Resource availability.
- Communication,
- Scope management,
- Customer requirements definition,
- Over-involvement by customer team members, and
- Lack of knowledge transfer to delivery team.

However, the challenges may show variability according to the experience of the manager, most of the experienced project managers may agree with the items given by the other one. Project managers may meet with one or more of these challenges during project completion in every industry. In order to overcome some of these challenges such as unrealistic deadlines, resource competition, uncertain dependencies, failure to manage risks and finish the project on time; a flexible to change, quick response, and easy to manage project scheduling methodology is needed.

BCO can produce good quality results to TSP which are very close to optimal values and in a reasonable computing time (Teodorovic, 2008). If BCO can produce better solutions to TSP than the available methods, it can be thought that BCO can also give good solutions for PSP. Therefore, the aim of the thesis is to solve the PSP with BCO.

CHAPTER III

BEE COLONY OPTIMIZATION

As well-known, the complex optimization problems is an area of research that needs improvement using new methods and approaches. Solving hard problems within acceptable levels of time and precision is a challenge and it is known that finding the optimal solution to complex multi-variable optimization problems within polynomial computation times is not possible (Pham et al, 2005; Pham et al, 2006a). Complex problems can be solved up to some degree using traditional engineering models, algorithms, heuristics and meta-heuristics (Teodorovic et. al., 2006). Methods which living creatures find to solve their problems for survival in nature are inspiring researchers in many different ways to develop models for their problems where optimization is one of the fields natural systems give acceptable solutions (Pham et al., 2005; Karaboga and Akay, 2010).

As stated before natural systems especially social insect colonies can direct the studier to create systems able to perform highly complex tasks by dynamically interacting with each other (Teodorovic et. al., 2006). The bee system is said to be one of these systems that can find promising food sources by communicating within the hive.

An intelligent optimization tool BCO is inspired by the food foraging behavior of honey bees (Pham et. al., 2006a). In nature bees are searching for food and carrying the information outside the hive inside to find the best quality nectar and the maximum amount to fill the hive. Similarly the algorithm is a search algorithm capable of locating good solutions efficiently. In literature this approach is being used to solve continuous optimization problems, training neural networks, mechanical and electronic components design optimization, combinatorial optimization problems such as job shop scheduling, internet server optimization problem, TSP, etc (Nakrani and Tovey, 2004).

The food foraging system of honey bees is explained by Pham et al. (2006a). According to them a colony of honey bees can extend itself over long distances (more than 10 km) and in multiple directions simultaneously to exploit a large number of food sources. The foraging colony sends scout bees to these flower patches. Scout bees move randomly from one patch to another. After returning to the hive the scout bees who found a promising patch go to the dance floor. On the dance floor scout bees perform a dance called waggle dance which gives information about the flower patch they found. The dance gives information about the quality rating, direction and distance of the flower patch. With this information the colony sends the bees to the flower patches. The more the rating of the flower patch is the more the follower bees are sent there. While harvesting from a patch, the bees monitor is food level. If the patch is still good enough as a food source then it will be advertised in the next waggle dance and more bees will be recruited to that source.

III.1. PARAMETERS OF BCO

The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. Table III.1 gives the pseudo code for the algorithm in its simplest form (Pham et. al., 2006a).

Table III.1. Pseudo code of the basic bee algorithm

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) // Forming new population.
a. Select sites for neighborhood search.
b. Recruit bees for selected sites (more bees for best e sites) and evaluate fitness's.
c. Select the fittest bee from each path.
d. Assign remaining bees to search randomly and evaluate fitness's.
4. End while.

While solving a problem with BCO, "n" solutions are chosen randomly. This means than each scout bee represent a randomly chosen solution to the related problem. "n" solutions are rated according to the objective function. The rating results with "e" very best solutions, and "m" best solutions of the first iteration. For "(e) + (m-e)" solutions neighborhood search is accomplished. Neighborhood search is respectively defined for the problem. The best solution found during neighbourhood search is passed on the next iteration, until the stopping criterion is met.

The results obtained on some benchmark show that BCO outperforms deterministic Simplex method, stochastic simulated annealing optimization procedure, GA, and Ant Colony System (ACS) in terms of speed of optimization and accuracy of the results (Pham et. al., 2006a).

III.2. INTRODUCTION OF BCO TO LITERATURE

Earlier research on bees where the researchers were impressed by the cooperative behavior of them to generate an algorithm was published by Yonezawa and Kikuchi (1996). They showed that this algorithm has been effective in speed of decision and finding solution of complex problems. Following them Sato and Hagiwara (1997) used Bee System (BS) to improve GA and proved that the bee system has been more effective especially for highly complex multivariate functions. After these early researches bees disappear from the researchers' standpoint for a six years period. Cox and Myerscough (2003) has studied on another theory and modeled the foraging of a honey bee colony. In their research they compared the performance of a heterogeneous colony to a homogeneous colony in foraging behavior. Attention on bees in optimization continued and Karaboğa (2005) published a report on behavior of honey bee swarm for numerical optimization. In his report he also described a new optimization algorithm based on the intelligent behavior of honey bee swarm.

Whereupon, the life style of bees has reached to its peak in optimization world after 2006, being introduced once again to literature by Pham et al. (2006a) and Teodorovic et al. (2006) separately and applied it to many cases. Karaboğa and Baştürk (2007) presented another theory based on the intelligent behavior of honey bee swarm named Artificial Bee Colony (ABC) Algorithm. Unlike BA, in ABC here

exist three types of bees: employed, onlooker and scout. In this system employed bees search flower patches, measure nectar amounts and carry the information to the onlookers who decide and send the scout bees to the food sources.

Table III.2. Theory Building Researches on BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Yonezawa & Kikuchi (1996)	Theory Building	Collective honey bee behavior	An algorithm being generated with the co-operative behavior of bees, being effective in speed of decision and solution of complex problems.
Sato & Hagiwara (1997)	Theory Building	Improving GA with BS	Improved GA is being obtained high ability for local search without degrading the global search ability, validity of the BS and BS being more effective especially for highly complex multivariate functions.
Cox & Myerscough (2003)	Theory Building	A model of foraging by a honey bee colony	A heterogeneous colony may perform better than a homogeneous colony in foraging behavior.
Pham et al. (2006a)	Theory Building	BA	BA is being introduced to literature and having remarkable robustness with 100% success in all cases of multi-modal functions in n-dimensions.
Teodorovic et al. (2006)	Theory Building / Case Study	BCO Principles and applications	BCO is being proposed; two algorithms being described: BS and Fuzzy Bee System (FBS); FBS is being capable of solving deterministic combinatorial problems.
Karaboğa & Baştürk (2007)	Theory Building / Comparative Study	ABC Algorithm	ABC is being based on the intelligent behavior of honey bee swarm, and used for optimizing multivariable functions, ABC outperforming GA, Particle Swarm Optimization (PSO), and Particle Swarm Inspired Evolutionary Algorithm (PSIEA).
Lemmens et al. (2007)	Comparative Study	Comparison of the non-pheromone based algorithm with a pheromone based algorithm in the domain of foraging	Non-pheromone based algorithms being more efficient when finding and collecting food with three times less iterations, less total computation time but being less adaptive than a pheromone based algorithm.
Quijano & Passino (2007a)	Theory Building	Honey bee social foraging algorithms (Part I)	Ideal free distribution concept from theoretical ecology is being introduced, and ideal free distribution being a global optimum point for both a single hive and multiple hives.
Lu & Zhou (2008)	Theory Building	Bee Collecting Pollen Algorithm	Performance of algorithm being comparable to Ant Colony Algorithms, but gapping behind the efficient TS heuristics.
Pham & Haj Darwish (2008)	Method Building	Fuzzy Selection of Local Search Sites in BA	A combination of BA as a core and a fuzzy system for the greedy selection is being applied for function optimization, and improved BA being robust and reliable for function optimization.

Lemmens et al. (2007) compared non-pheromone based algorithm with the pheromone based one in the domain of foraging and found out once more that non-pheromone based algorithm is being more efficient.

Ideal free distribution concept from theoretical ecology has been introduced by Quijano and Passino (2007a) under the name Honey Bee Social Foraging Algorithms. It is found that this distribution is being a global optimum for both a single and multiple hives. In the second part of their research engineering applications of Honey Bee Social Foraging Algorithm is provided.

Lu and Zhou (2008) also researched bee system under the name Bee Collecting Pollen Algorithm with respective performance testing. It is found that performance of the algorithm is being comparable to Ant Colony Algorithms but gapping behind the efficient TS Heuristics.

Pham and Haj Darwish (2008) researched on improving the local search sites of BA. In order to improve BA, they combined BA as a core and a fuzzy system for the greedy selection. This method is applied to a set of benchmark function optimization problems. The results showed the robustness and reliability of the BA.

In Table III.2 a summary of theory building researches on BCO can be found.

III.3. APPLICATIONS OF BCO IN LITERATURE

After the introduction of BCO, case studies and comparative studies appear in literature continually. BCO has been applied to several cases on training of neural networks, scheduling, TSP, data clustering, manufacturing cell formation, robotics, p-center problem, and large scale numerical problems, stochastic problems, multi-objective optimization, etc.

The BCO algorithm's first application on scheduling problems is job shop scheduling case. In a comparative study performed by Chong et al. (2006), the performance of BCO in jobshop scheduling is compared to ACO and TS Heuristics. Another scheduling research for BCO is investigated by Pham et al, (2007d) with the case of scheduling jobs for a single machine. Pham and Koç (2008) have also published the Flowshop Sequencing tasks with a set of benchmark problems. Wong et al. improved BCO algorithm with Big Valley Exploitation and compared the performance of the method for job shop scheduling (2008c).

Table III.3 briefly summarizes the BCO case studies on scheduling domain.

Table III.3. Scheduling Applications with BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Chong et al. (2006)	Case Study / Comparative Study	BCO algorithm to job shop scheduling	The performance of BCO algorithm being comparable to ACO algorithms but gapping behind the efficient TS heuristics.
Pham et al. (2007d)	Case Study	Scheduling jobs for a machine using BA	Being first to report application of BA to a combinatorial optimization problem, performing more strongly than existing techniques, BA being more stable and robust than other algorithms.
Pham & Koç (2008)	Case Study	Flowshop Sequencing with BA	BA outperforming other techniques in terms of the efficiency and robustness.
Wong et al. (2008c)	Comparative Study	BCO Algorithm with Big Valley Exploitation	Bee Colony Big Valley is being able to achieve comparable performance to Tabu Search Algorithm, Shifting Bottleneck Heuristic and Bee Colony Algorithm with Neighbourhood Search using small number of iterations with better performance of computation time.

The case studies by Pham et al. (2006b, 2006c, 2006d, 2006e) were on utilizing BCO for training support to neural networks with different cases. These cases include the application of BA to the training of Radial Basis Function (RBF) networks for the recognition of control chart patterns (Pham et al., 2006b), optimization of weights of multi-layered perceptrons using BA (Pham et al., 2006c), application of BA to the learning vector quantization networks for recognition of control chart patterns (Pham et al., 2006d), and optimization of neural networks for wood defect detection using BA (Pham et al., 2006e).

Pham and Sholedolu (2008) also used a hybrid PSO-BA to train neural networks for wood defect classification. This combination trained classifiers more accurately than the original PSO and Backpropagation method (Pham and Sholedolu, 2008).

One other BCO supported neural network application was studied by Öztürk and Karaboğa (2008) for classification and clustering problems. The performance of ABC is tested for classification of data sets: cancer, diabetes and heart. Pham and Haj Darwish (2010) have improved BCO with Kalman Filtering to train neural networks for pattern classification which were able to identify defects more accurately than original PSO and the backpropagation method.

Table III.4 provides the summary of neural network applications with BCO.

Table III.4. Neural Network Applications with BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Pham et al. (2006b)	Case Study / Comparative Study	Application of BA to the training of RBF networks for the recognition of control chart patterns	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers, BA producing RBF networks with a lower error value than that obtained for an RBF network trained conventionally.
Pham et al. (2006c)	Case Study / Comparative Study	Optimization of weights of multi-layered perceptrons using BA	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers than the backpropagation algorithm.
Pham et al. (2006d)	Case Study / Comparative Study	Application of BA to the training of learning vector quantization networks for the recognition of control chart patterns	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers than the standard learning vector quantization training algorithm.
Pham et al. (2006e)	Case Study / Comparative Study	Optimization of neural networks for wood defect detection using BA	The accuracy obtained by the BA being comparable to backpropagation algorithm, and BA being faster.
Pham & Sholedolu (2008)	Method Building	Combining BA with PSO to train Neural Networks for Wood Defect Classifier	Despite the high dimensionality of the problem, the PSO-BA trained classifiers being able to identify defects more accurately than original PSO algorithm and backpropagation method.
Ozturk & Karaboga (2008)	Case Study	Classification by neural networks and clustering with ABC Algorithm	The performance of ABC algorithm being tested on training of Artificial Neural Networks and clustering for classification of different data sets of cancer, diabetes and heart where ABC algorithm being successfully able to be applied.
Pham & Darwish (2010)	Methodology / Case Study	BA with Kalman Filtering to train artificial neural network for pattern classification	Combining random exploration and greedy exploitation, giving better test results, reducing number of iterations and speeding up the training manifold.

BCO has also achieved competing performance than the available methods in most of the robotics applications. In the cases of tuning a fuzzy logic controller for a robot gymnast (Pham et al., 2007c), learning the inverse kinematics of a robot manipulator (Pham et al., 2008a), and the design of a Proportional Integral Derivative (PID) controller (Pham et al., 2008b); BCO outperformed the existing methods. Karaboğa and Akay (2010) compared the effect of problem dimensionality on the performance of ABC, Harmony Search (HS) and BA on time-delayed and ossilatory PID control systems. However, Karaboğa and Akay's research concluded that ABC showed a slightly better performance than BA and Harmony Search.

Table III.5 provides the robotics applications with BCO.

Table III.5. Robotics Applications with BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Pham et al. (2007c)	Case Study	Tuning a fuzzy logic controller for a robot gymnast using BA	Controller tuned with BA giving a smooth performance with fewer variations and smaller input control signals than in the case of the unturned controller, BA being a useful tool for tuning fuzzy logic controllers to achieve better performance.
Pham et al. (2008c)	Case Study / Comparative Study	Learning the inverse kinematics of a robot manipulator using BA, training multi-layer perceptron neural networks to model the inverse kinematics of n articulated robot manipulator arm	BA producing accurate learning results for the three modeling tasks, high reliability of BA, superior performance of BA over Backpropagation Algorithm and Evolutionary Algorithm.
Pham et al. (2008d)	Case Study	Design of PID controller for a flexible single-link robot manipulator using BA	PID controller which is tuned with BA to minimize vibrations and positioning errors being both efficient and robust.
Karaboga & Akay (2010)	Case Study / Comparative Study	The effect of problem dimensionality on the performance of basic ABC, HS and BA	ABC being showed a slightly better performance on time-delayed and oscillatory PID control systems.

TSP applications of BCO have been researched by Wong et al. (2008a, 2008b). In the former research (Wong et al., 2008a) a BCO based model is used, however, the following research (Wong et al., 2008b) integrated 2-opt heuristic with BCO improving its performance. Table III.6 summarizes TSP applications with BCO.

Table III.6. TSP Applications with BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Wong et al. (2008a)	Model Building / Comparative Study	BCO for Traveling Salesman Problem (TSP)	A BCO based model being introduced to solve TSP, and model being tested on a set of benchmark problems.
Wong et al. (2008b)	Model Building / Comparative Study	BCO with Local Search for TSP	BCO being integrated with 2-opt heuristic to solve TSP, integration improving BCO performance significantly.

Other applications with BCO include Manufacturing Cell Formation (MCF) problem (Pham et al., 2007a), multi-objective optimization (Pham et al, 2007b), optimization of a Support Vector Machine (SVM) (Pham et al, 2007e), data clustering (Pham et al, 2007f), engineering applications (Qijano and Passino, 2007b), large scale numerical problems with ABC (Karaboğa and Akay, 2008), fuzzy clustering (Pham et al., 2008a), generating branded product concepts (Pham et al., 2008b), multi-objective environmental/economic power dispatch (Pham et al.,

2008e), a stochastic optimization problem (Pham et al., 2008f), ride-matching problem (Teodorovic and Dell'Orco, 2008), p-center problem (Davidovic et al., 2010).

Table III.7 provides information about other applications with BCO.

Table III.7. Other applications with BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Pham et al. (2007a)	Case Study	MCF with BA	BA being very effective in generating optimal solutions to the MCF problem.
Pham et al. (2007b)	Case Study	Multi-Objective optimization with BA	A modified version of the BA, BA being computationally fast multi-objective optimizer tool.
Pham et al. (2007e)	Case Study	Optimization of a SVM for wood defect classifier using BA, best combination of SVM parameters and data features to maximize defect classification accuracy	The accuracies obtained being higher than that of obtained by a standard support vector machine, reconfirming usefulness of BA as an optimization tool.
Pham et al. (2007f)	Case Study / Method Building	Data Clustering using BA	Proposing a clustering method that integrates the simplicity of the k-means algorithm with the capability of BA to avoid local optima, the proposed method not being trapped at locally optimal solutions, and producing better performances than those of the k-means algorithm and the GA-Clustering method.
Quijano & Passino (2007b)	Case Study	Honey bee social foraging algorithms (Part II)	Engineering application on honey bee social foraging algorithm.
Karaboğa & Akay (2008)	Theory Building / Comparative Study	Large scale numerical problems with ABC Algorithm	The performance of ABC algorithm being compared to Differential Evolution and PSO for large scale numerical problems, and ABC algorithm having superior performance over Differential Evolution and PSO.
Phamet et al . (2008a)	Case Study / Comparative Study	Fuzzy Clustering with BA	BA combination with the Fuzzy C-Means Algorithm improving its clustering results significantly and giving better results than the Fuzzy C-Means Algorithm combined with GA.
Pham et al. (2008b)	Case Study / Comparative Study	Generating product design concepts using BA	Comparison of BA with Evolutionary Algorithm for the task of maintaining the brand identity of the product, an application on a new Coca Cola Bottle design, and BA outperforming the Evolutionary Algorithm.
Pham et al. (2008e)	Case Study / Comparative Study	Multi-objective Optimization of Environmental/Economic Power Dispatch with BA	Minimization of fuel cost and nitrogen oxide emissions simultaneously using BA with Pareto optimality, comparison with other methods, BA being suitable for the problem.

Table III.7. Other applications with BCO (continues...)

Author(s) & Year	Methodology	Research Area	Major Findings
Pham et al. (2008f)	Case Study	Solving A Stochastic Optimization Problem with BA	BA is being used to determine the parameter setting for an ACS to find the shortest route of a TSP, and parameter setting produced by BA outperforming other methods.
Teodorovic and Dell' Orco (2008)	Method Building	Solving the ride-matching problem by BCO	No theoretical rules supporting such a proposed approach, but BCO appearing very promising.
Davidovic et al. (2010)	Case Study / Comparative Study	BCO for the p-center problem	A variant of BCO based on the improvement concept (BCOi) is proposed, BCOi being a very useful concept for solving difficult combinatorial problems, and generating high-quality solutions.

The research on BCO is provided in Appendix I – Table 1 in chronological order.

CHAPTER IV

PROJECT SCHEDULING WITH BEE COLONY OPTIMIZATION ALGORITHM (A PROPOSED MODEL)

IV.1. PROJECT SCHEDULING WITH BEES

As stated earlier project scheduling is a complex activity requiring computationally effective solutions. BCO algorithm is considered to be an alternative for this as proven to be able to cope with the respective computational complexity to some degree. In this study the possibility of utilizing this algorithm for project scheduling is investigated and a respective model is proposed. This section provides problem mapping and allocations mechanism which will be the baseline for the proposed model. In the first section the problem mapping for project scheduling using bees is introduced. In the second section forager allocation mechanism is explained and following this the main model is described.

IV.2. DEFINITIONS

This section defines the project scheduling problem using the parameters of BCO algorithm. The transformation of the project scheduling problem into bee foraging problem is called "problem mapping". With problem mapping project scheduling problem is transferred into a bee foraging problem.

The PSP researched here has multi-modes which mean that each activity can be processed with different resources. Each mode refers to an activity completion time with a different resource having different speed or duration for that activity. From this point of view a scout bee carries the information for the single activity-resource combination to the hive. In this manner a mode is chosen for every activity in the project. Mode carries the information of the resource and the duration of processing the activity on that resource. In addition with the precedence constraints a completion time can be calculated for the project. Finally, a scout bee stands for the completion time of the project for the chosen single resource-activity combination. The objective

of the problem is to minimize the completion time of the project regarding the resource allocations.

There will be (n) scout bees in the solution space searching for a suitable path combination leading to the best solution available in the project pool. The project pool including alternative solutions is the solution space of the respective scheduling problem. Similar to the visited sites in foraging problem, (n) will also indicate the number of varying paths (equal to number of scout bees) using alternative resources with different durations or having different costs for the specified tasks in the schedule.

Using the same kind of analogy, (m) will be the number of solutions selected out of n different path combinations whereas (e) will stand for the number of best solutions chosen among the m solutions.

The number of bees recruited for best (e) sites is referred as (nep) representing the number of sites to be searched around the best (e) solutions in BA. Sites are the flower patches around in bee foraging problem, however in PSP site refers to completion times around the chosen resource-activity combination. Each site is gained by changing the resource of an activity on the critical path. Similarly (nsp) will denote the number of sites to be searched around the other (m-e) solutions.

IV.3. FORAGER ALLOCATION MECHANISM FOR PROJECT SCHEDULING PROBLEM

For bees solution space is the flower patches whereas in PSP solution space in the project pool. The project pool consists of every completion time for each resource-activiy combination. Similar like the bees forage for food in flower patches, in order to seek for the best resource-activity combination giving the shortest completion time; a resource is chosen randomly for each activity. The completion times for "n" combinations are calculated and evaluated. For the promising solutions a neighborhood search is performed. This is provided by changing a resource for a chosen activity randomly in PSP. After evaluating the solutions around, the best solution in the neighborhood is carried to the next iteration until the stopping criterion is met.

The foraging allocation mechanism for PSP is described in Table IV.1 and is adopted from the pseudo code of Pham et. al., (2006a).

Table IV.1. Forager Allocation Mechanism for PSP

1. Initialize project pool with random solutions.
2. Evaluate the fitness of the project pool.
3. While (stopping criterion not met) // Forming new project pool.
a. Select solutions for neighborhood search.
b. Allocate number of search solutions for selected sites (more searching area for best e solutions).
c. Select the fittest resource-activity combination from each project.
d. Assign remaining resource-activity combinations to search randomly and evaluate their fitness.
4. End while.

IV.4. MULTI – MODE PROJECT SCHEDULING PROBLEMS

In order to understand why BA is used in this study, multi-mode project scheduling problems are defined in this section. In these kinds of problems there are alternative resources for each activity of the project, where each activity can be completed using many one of the alternatives given. This provides alternative project completion times with different resources combinations. As far as resources and/or the way activities are executed are not identical, the duration of each mode per activity will be different from others. Somehow, finding the shortest completion time of the project depends on the selection of resources per activity and the sequence of the activities on the resources.

As well known in project scheduling, the shortest completion time of the project is the resembled by the longest path duration from the starting node till the ending one. That is to say that, in this case, to find out the optimum project completion time, each alternative path has to be traced in such a way that the longest path will be the shortest project completion time for each alternative resource combination. After finding the project completion time for each alternative resource combination, the resources to be used in the project are set up. This process presents a challenge especially in large scale projects.

In order to understand the solution space size for large scale project, the number of alternatives to schedule a project is calculated with the following formula.

r_i : number of resources for activity i,

k : the number of activities to be completed,

S : solution space of alternative completion ways for the project},

$n(S) = (r_1 * r_2 * \ldots * r_i * r_{i+1} * \ldots * r_k)$

For example, if there are 4 activities (jobs) in a project where job 1 can be completed using one of the 4 resources, job 2 can be completed using one of the 2 resources, job 3 can be completed using one of the 3 resources, job 4 can be completed using one of the 5 resources; then we there will be $4 * 2 * 3 * 5 = 120$ alternative ways to complete the project.

Among the alternative ways, the shortest completion time will be sought. The list of alternative solutions is considered to be the solution space (project pool) where the bees will scout for the best solution. As the number of project activities and modes to perform these activities increase, the project pool enlarges rapidly. After some dimension finding the shortest completion time of the project, becomes as hard as finding a predefined blob in an olympic pool. In this instance problem becomes np-hard for large scale problems where the number of project completion alternatives happen to be so called inaccessible with manual techniques. The nature of BA has the ability to evaluate the project pool at different random points. Accessing to different points at the same time accelerate reaching a suitable (possibly the best) solution, and the algorithm sweeps the project pool by eliminating the ineffective solutions and tracing around the better ones to converge towards the best solution.

IV.5. A PROJECT SCHEDULING PROBLEM

In order to understand the applicability of BCO algorithm to project scheduling, a sample case problem is created as illustrated in Table IV.2. For the sake of simplicity, the size of the problem is intentionally kept to a small number of activities. There are eight activities and two modes in the sample problem. In Table IV.2 the given first column is the activity number; second column provided shows the alternative resources for each activity. There are a total of eight resources available; however each activity can be performed with separately either one of the two resources. The third and fourth columns show the preceding and following activities per each activity respectively.

Table IV.2. Activity Information for Sample PSP

Activity No	Available Resources	Preceding Activity	Following Activity
1	R1, R2	-	2, 3
2	R3, R5	1	4, 5
3	R6, R7	1	5
4	R4, R8	2	6
5	R2, R7	2, 3	6, 7
6	R1, R8	4, 5	8
7	R2, R3	5	8
8	R6, R4	6, 7	-

Using the information provided in Table IV.2, a project network can be drawn as shown in Figure IV.1.

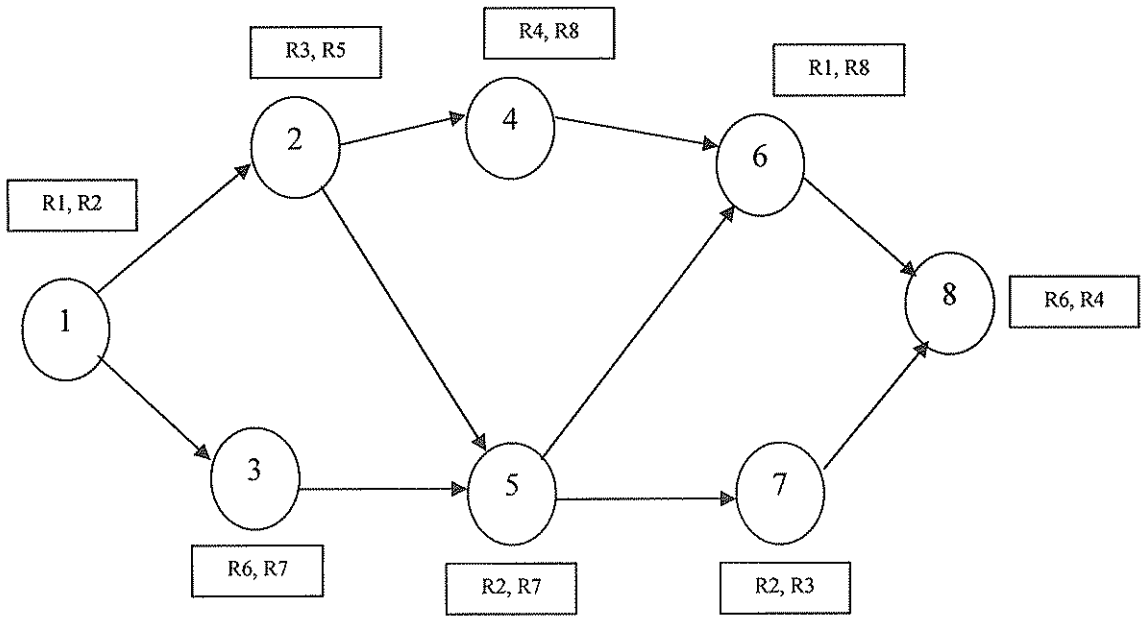

Figure IV.1. Project Network of the Sample Problem

Job durations are needed in order to calculate the project completion time. In the Sample Problem, job durations for each job on an alternative resource is different. Therefore, in order to calculate the completion time of the alternative schedules, activity times should be known. Table IV.3 indicates activity times for the appropriate resources available per each activity.

Table IV.3. Job durations for resources

Activity	Resource	Duration	Resource	Duration
1	R1	t_{11}= 5	R2	t_{12}=3
2	R3	t_{23}=1	R5	t_{25}=6
3	R6	t_{36}=8	R7	t_{37}=3
4	R4	t_{44}=9	R8	t_{48}=4
5	R2	t_{52}=5	R7	t_{57}=3
6	R1	t_{61}=7	R8	t_{68}=2
7	R2	t_{72}=4	R3	t_{73}=2
8	R6	t_{86}=6	R4	t_{84}=4

Setting up a time frame for each activity can make it possible to calculate the project completion time. Note that, in this Sample Project there are 5 paths to complete the overall project as such;

- 1-2-4-6-8,
- 1-2-5-6-8,
- 1-2-5-7-8,
- 1-3-5-6-8,
- 1-3-5-7-8.

Note that, in the Sample Problem, each job has 2 alternative resources; this in turn creates 2^8 = 256 alternative solutions. From these, the shortest completion time will be sought. This list of alternative solutions is considered to be the solution space (project pool) where the bees will scout for the best solution. Among the alternatives "n" randomly chosen solutions are ranked. After listing the "n" solutions in ascending order, first "m" solutions are evaluated and the better "e" solutions are separated. More neighborhood search is accomplished around "e" and less around "m-e" solutions. During neighborhood search the resources of the activities in critical path are changed randomly.

For the stopping criterion, a number of iterations are chosen for the algorithm. The algorithm stops when the number of iterations is carried out. If the number of iterations is high enough the system converges to the solution. The time computed depends on the size of the solution space and the number of iterations. The list of the solution space where the bees scout for the best solution in the Sample Problem is

given in the Appendix 2. It includes the resource alternative for each job (given as node). For each five paths to be traced project completion is calculated. In the last column project completion time for each alternative combination is chosen among the five paths. Table IV.4 provides a part of the project pool.

Table IV.4. A part of the project pool

	Node1		...		Node8		Path Completions					
	Res.	Time	...	...	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
1	R1	5	...	...	R6	6	17	13	10	20	17	**31**
2	R2	3	...	...	R6	6	17	13	10	20	17	**29**
3	R1	5	...	...	R6	6	22	18	15	20	17	**33**
4	R2	3	...	...	R6	6	22	18	15	20	17	**31**
5	R1	5	...	...	R6	6	17	13	10	15	12	**28**
...	...	...	...	...	...	...	...	...	...	...	...	**...**

IV.6. UTILIZATION OF BA FOR PROJECT SCHEDULING PROBLEM

In this section, the proposed BCO based scheduling model is explained step by step. Keep in mind that the project pool consists of alternative solutions (flower patches in foraging); the initial size of the patches (ngh) can be defined as the quantity of solutions in the project pool. In the Sample Problem this is set to ngh=256.

Other parameters (n, m, e, nep, and nsp) are defined as the following. Because the solution space is 256 the parameters can be small values. The number of visited sites (n) is chosen as 10, the number of selected sites (m) is set to 2 and the number of best sites among the (m) selected sites (e) is considered to be equal to 1. Similarly, the number of bees recruited for best (e) sites (nep) is equal to 10 and the number of bees recruited for the other (m-e) sites (nsp) is set to 4. The stopping criterion, the total number of iterations is selected 10.

Following the determination of the values of the parameters; the next step is to initialize the project pool with random solutions. Using a random generator, 10 random solutions among the solutions in the project pool are selected for this purpose. Among those, the best 2 of them (m) are selected and the one very best (e) solution is identified. Suppose the 2 of the best solutions (with respect to completion time) are;

- $m_1 \rightarrow 20$ time unit
- $m_2 \rightarrow 22$ time unit

Note that, since the best completion time among these at this moment is m_1=20, it is set as the best solution (e) of the project pool. These solutions were considered to be the focal point to search better solutions around. A neighborhood search step for the algorithm is used for this purpose. During this search (nep) = 10 around the (e) = 1 best site and (nsp) = 4 around the remaining (m – e) = 1 site are sought. Table IV.5 gives the nep solutions for the 1st iteration.

Table IV.5. The "**nep**" Solutions for the 1st Iteration

	Node 1		...		Node 8		Path Combinations					
	Res.	Time	...	...	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	Cmax
188	R2	3	...	...	R4	4	12	11	13	13	15	**22**
189	R1	5	...	...	R4	4	7	6	8	8	10	**19**
190	R2	3	...	...	R4	4	7	6	8	8	10	**17**
191	R1	5	...	...	R4	4	12	11	13	8	10	**22**
192	R2	3	...	...	R4	4	12	11	13	8	10	**20**
193	R1	5	...	...	R4	4	17	13	8	20	15	**29**
194	R2	3	...	...	R4	4	17	13	8	20	15	**27**
195	R1	5	...	...	R4	4	22	18	13	20	15	**31**
196	R2	3	...	...	R4	4	22	18	13	20	15	**29**

After the neighborhood search around m_1=e; a better solution is found. This time another search will be carried out around m_2. However it will contain less solution. Table IV.6 provides the nsp solutions for the 1st iteration.

Table IV.6. The "**nsp**" Solutions for the 1st Iteration

	Node 1		...		Node 8		Path Combinations					
	Res.	Time	...	...	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	Cmax
239	R1	5	...	...	R4	4	12	13	13	10	10	**22**
240	R2	3	...	...	R4	4	12	13	13	10	10	**20**
241	R1	5	...	...	R4	4	12	6	6	13	13	**22**
242	R2	3	...	...	R4	4	12	6	6	13	13	**20**
243	R1	5	...	...	R4	4	17	11	11	13	13	**26**

The second search yields two solutions resulting 20 time unit. Yet, this is not better than best solution around e in this iteration. So that, 17 is carried to the next iteration as n_1; one of the next iteration's random solution. A random search is sought again for the second iteration. This iteration ends with a best solution of 15 time units. If a better solution than the former one exists, the latter takes the place of the previous best solution. In this case 15 take the place of 17 in the 2nd iteration. And this procedure is repeated 10 times; a better solution than 15 cannot be provided. The algorithm ends finding a best solution with project completion time 15 for the

Sample Project, yielding the respective resource combination for each activity identified in Table IV.7.

Table IV.7. Best resource utilization for the sample problem

Activity	Resources	Duration
1	R2	$t_{12}=3$
2	R3	$t_{23}=1$
3	R7	$t_{37}=3$
4	R8	$t_{48}=4$
5	R7	$t_{57}=3$
6	R8	$t_{68}=2$
7	R3	$t_{73}=2$
8	R4	$t_{84}=4$
Completion time: **15** units		

Using the information provided in Table IV.7 the project network can be drawn as in Figure IV.2 with the critical part given in bold fonts.

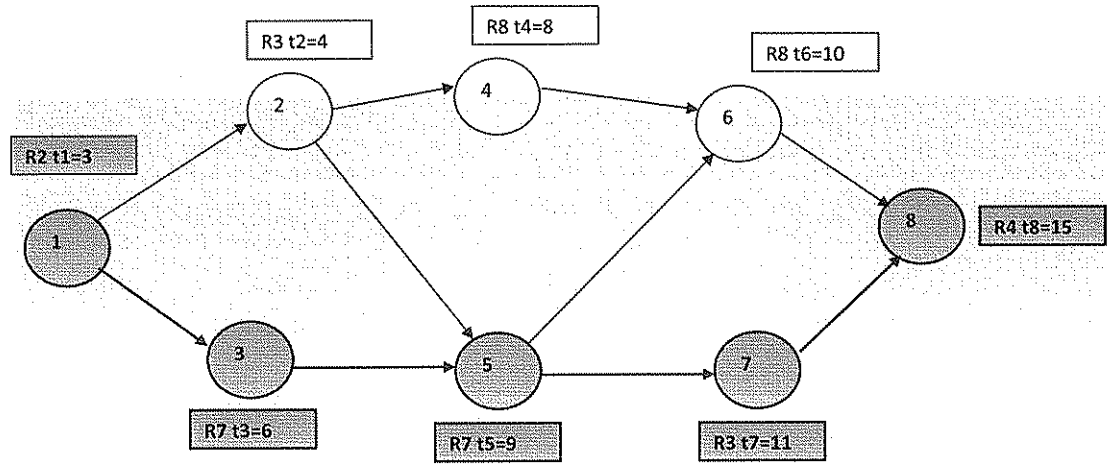

Figure IV. 2. Network of Best Resource Utilization for the Sample Project

CHAPTER V

REAL-TIME PROJECT TRACKING

Even if a project scheduling model using BCO is defined, it is not sufficient for tracking of a project. In order to check if the project is being realized in accordance with pre-defined schedule, an appropriate tracking system should be defined. As projects are realized in real-time and sudden changes along the project may affect the project flow critically, a real-time tracking system for the scheduler giving quick response to these sudden changes seems vital for better project management. In this section, first the information about the real-time scheduling is given and then the real-time project tracking model is introduced.

V.1.REAL-TIME SYSTEMS

Real-time system requiring deterministic or probabilistic timing; historically, were scheduled by cyclic executives whereas during 1970's and 1980's, researchers realized that the systems produced by this static approach were inflexible and difficult to maintain (Sha et al., 2004). They reported that under the leadership of Andre van Tilborg, this understanding led to the launch of a Real –Time Systems initiative by the United States Office of Naval Research.

Mohammadi and Akl (2005) define real-time systems consisting of a controlling system (computer) and a controlled system (environment). In this aspect, the controlling system interacts with its environment based on information available about the environment. According to them a real-time system processes as within the given conditions and approach:

- A real-time computer contains sensors which provide readings at periodic intervals in order to control a device or process.
- The computer sends signals to actuators for response.
- The unexpected or irregular events must also receive a response.
- The response should be delivered within a time bound.

- The computer meets these demands depending on its capacity to perform the necessary computations in the given time.
- If a number of events occur close together, either the computer will schedule the computations or the system will be unable to meet all the possible unexpected demands. In this situation the system is said to lack sufficient resources.
- The outcomes of failure to meet the timing constraint for a response can be different such as
    - there may be no effect at all,
    - the effects may be minor or correctable,
    - the results may be catastrophic.
- Timing properties of each task should therefore be considered when scheduling tasks on a real-time system.

Multiple tasks with different levels of criticality constitute real-time applications normally and a system is said to be real-time if it contains at least one task, which falls into one of the following categories (Mohammadi and Akl, 2005):

- **Hard real-time** task where the execution of the task should be completed by a given deadline. Hard real-time tasks cannot miss any deadline; otherwise, undesirable or fatal results will be produced in the system.
- **Soft real-time** task where even if soft real-time tasks miss deadlines, the system can still work correctly. In spite of the fact that in a real-time system, missing deadlines is an undesirable situation, in some cases a penalty is paid for the lateness computation after a given deadline. The later the task finishes, the more penalty is paid.
- **Firm real-time** task where the earlier the task finishes its computation before a given deadline, the more rewards it gains.

V.2. REAL-TIME SCHEDULING

Basically, in a scheduling problem given a set of jobs are executed before the overall deadline in a sequence under the various constraints such as resource availability, precedence relations and deadlines. Anyhow the real-time scheduling problem is characterized with the properties of real-time scheduling tasks and the

goals which are considered when designing the system. Mohammadi and Akl (2005) defined the desired goals as:

- Meeting the timing constraints of the system,
- Preventing simultaneous access to shared resources and devices,
- Although attaining a high degree of utilization while satisfying the timing constraints of the system is important; this is not a primary driver,
- Reducing the cost of context switches caused by preemption,
- Reducing the communication cost in real-time distributed systems; possibility of finding the optimal way to decompose the real-time application into smaller portions in order to have the minimum communication cost between mutual portions where each portion is assigned to a computer.

In addition, they also emphasized that the following items are also desired in advanced real-time systems:

- Considering a combination of hard, firm, and soft real-time activities, which implies the possibility of applying dynamic scheduling policies that respect the optimality criteria,
- Task scheduling for a real-time system whose behavior is dynamically adaptive, reconfigurable, reflexive and intelligent, and
- Covering reliability, security, and safety.

Based on these information they characterized the following properties appropriate scheduling methods for the real-time system:

Soft/Hard/Firm real-time tasks

The real-time tasks are classified as hard, soft and firm real-time tasks. This is already described in Section V.2.

Periodic/Aperiodic/Sporadic tasks

Periodic tasks are real-time tasks which are activated (released) regularly at fixed rates (periods). Normally, periodic tasks have constraints which indicate that instances of them must execute once per period.

Aperiodic tasks are real-time tasks which are activated irregularly at some unknown and possibly unbounded rate. The time constraint is usually a deadline.

Sporadic tasks are real-time tasks which are activated irregularly with some known bounded rate. The bounded rate is characterized by a minimum inter-arrival

period, that is, a minimum interval of time between two successive activations. The time constraint is usually a deadline.

An aperiodic task has a deadline by which it must start or finish, or it may have a constraint on both start and finish times. In the case of a periodic task, a period means once per period or exactly units apart. A majority of sensory processing is periodic in nature.

Preemptive/Non-preemptive tasks

In some real-time scheduling algorithms, a task can be preempted if another task of higher priority becomes ready. In contrast, the execution of a non-preemptive task should be completed without interruption, once it is started.

Multiprocessor/Single processor systems

The number of the available processors is one of the main factors in deciding how to schedule a real-time system. In multiprocessor real-time systems, the scheduling algorithms should prevent simultaneous access to shared resources and devices. Additionally, the best strategy to reduce the communication cost should be provided.

Fixed/Dynamic priority tasks

In priority driven scheduling, a priority is assigned to each task. Assigning the priorities can be done statically or dynamically while the system is running.

Flexible/Static systems

For scheduling a real-time system, we need to have enough information, such as deadline, minimum delay, maximum delay, run-time, and worst case execution time of each task. A majority of systems assume that much of this information is available a priori and, hence, are based on static design. However, some of the real-time systems are designed to be dynamic and flexible.

Independent/Dependent tasks

Given a real-time system, a task that is going to start execution may require to receive the information provided by another task of the system. Therefore, execution of a task should be started after finishing the execution of the other task. This is the concept of dependency. The dependent tasks use shared memory or communicate data to transfer the information generated by one task and required by the other one. While we decide about scheduling of a real-time system containing some dependent tasks, we should consider the order of the starting and finishing time of the tasks.

For the PSP researched in this study, it is vital to select the properties of the system and the tasks, in order to design the reasonable tracking system. The tasks of Sample PSPs are soft real-time tasks, in which a penalty is paid lateness per unit time and aperiodic on the grounds that each task in a project might be different from each other. The preemption of the tasks is not allowed unless an interruption occurs in the project environment. For instance if a resource becomes unavailable permanently or temporarily then the task can be preempted and processed on another resource.

It is stated before that in single processor systems the tasks in the process on a single processor; however, in multi processor systems the tasks execute on multi processors simultaneously (Mohammadi and Akl, 2005). The processors in real-time systems can be corresponded to resources in PSP; where a task can be completed in either a single or multi resource environment.

When it comes to priority of the tasks, the most prior tasks in PSP are the ones on the critical path. It is the one in the project network which has the longest path from start node to sinking node. However, the PSP researched in this study has dynamic critical paths. Since project completion is calculated through the randomly chosen resource activity combinations, the critical path becomes dynamic with the randomness of the project completion. This situation generates dynamic priority tasks on the critical path which creates the project completion time.

For the flexibility of the system, it has a static due date which means that it cannot be changed but if a delay occurs in the system, then a penalty occurs. And finally, the task of a project cannot be started before its predecessors are completed. With the addition of this rule the real-time system's tasks exist as dependent tasks which are connected through a project network. The Sample PSPs obtain these properties for the tasks and the real-time tracking model is constructed under these conditions.

V.3. REAL-TIME TRACKING OF THE SAMPLE PROJECT

In section IV.6 the steps of the BA for PSP is defined and an example problem is solved giving the schedule of resource combination with the minimum completion time.

In the second phase of the model, the project is to be tracked until it is completed. During project flow there is always the possibility of interruptions. These

interruptions may occur as the temporary or permanent interruptions of resources including

- the periodic maintenance of a resource,
- breakdown of a resource,
- change in the cost or duration of a resource, etc.

In the Sample PSP, there were 256 alternatives in the solution space assume a situation where the project flows as scheduled until time t=4. However, at time t=4 resource R7 has a breakdown and is being fixed. The tasks completed up to that time are jobs 1 and 2. Job 3 was being processed on R7 when it broke down. Suppose up to t=4, 33% of job 3 is completed. In this case there are two alternatives for completing job 3, the first one is to use resource R6 where the total of job 3 can be processed in 8 unit time. Since the 66% of job 3 remains, resource R6 spends 5.3 unit time to complete this job and job 3 finishes at time t=9.3.

The other alternative for this job is to wait until R7 is fixed. R7 will be fixed in 6 unit time and able to be used at time t=10. In this case job 3 will be completed at time t=10+2=12 which will consequently delay job 5, 6 time units. Since the other job using R7 is job 5, there are also two alternative for this job. One of them is waiting until R7 is fixed at time t=10 and be completed in 3 unit time, or using R2 with a start of time t=9.3 at point which its predecessor job 3 ends at R6 and be completed in 5 unit time.

In this new situation different opportunities are available to complete the project. Each situation given above can be evaluated as different solutions in the BA. When the resource alternatives for the other uncompleted jobs are considered, the new solution space decreases to 32 alternatives. After the modification the new situation of the project is rescheduled using the BA. After determination of the new solution for remaining jobs, this new schedule is combined with the completed jobs before the breakdown of the resource R7. Note that the project would be completed in 15 unit time before the breakdown, but after the resource R7 has been broken down the project could be completed in 19 units. However, the reschedule creates two alternative solutions with a completion time of 19 units. Job 4 isn't in the critical path, and the two alternatives of processing this job do not extend the project completion. Both of the alternatives that finish job 4 end up with the same project

duration. The resource combination for each situation is given in Tables V.1 and V.2. In the Table V.I. job 4 is processed by R8 with 4 unit time. In the Table V.2 job 4 is processed by R4 with 9 unit time.

Table V.1. 1st Resource utilization after generating a reschedule

Job	Resources	Duration
1	R2	t_{12}=3
2	R3	t_{23}=1
3	R7,R6	t_{37}+t_{36}=9.3
4	R8	t_{48}=4
5	R7	t_{57}=3,7
6	R8	t_{68}=2
7	R3	t_{73}=2
8	R4	t_{84}=4
Completion time: **19** units		

Table V.2. 2nd Resource utilization after generating a reschedule

Job	Resources	Duration
1	R2	t_{12}=3
2	R3	t_{23}=1
3	R7,R6	t_{37}+t_{36}=9.3
4	R4	t_{48}=9
5	R7	t_{57}=3,7
6	R8	t_{68}=2
7	R3	t_{73}=2
8	R4	t_{84}=4
Completion time: **19** units		

The project network, after rescheduling is provided in the Figure V.1.

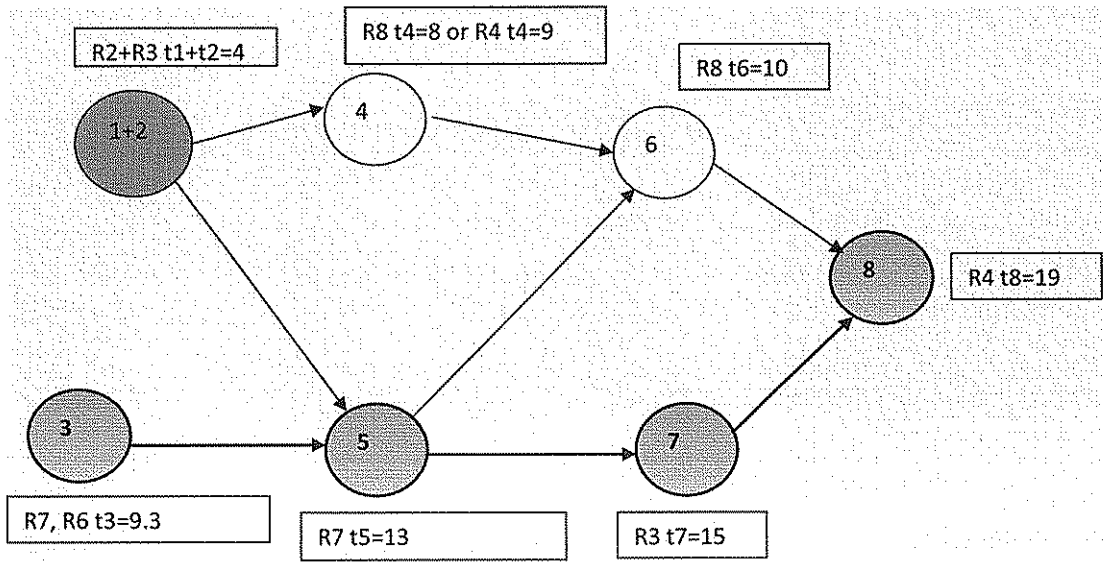

Figure V.1. Project schedule network after generating a reschedule

CHAPTER VI

BEE-BASED REAL-TIME SCHEDULING MODEL

Proposed Bee Colony Based Scheduling Model explained in Chapter IV is extended and improved as explained below. The model is redesigned in two fold; schedule generation and real-time tracking. First the scheduler chooses one of the resources for each task to be accomplished. BA is utilized for this purpose. The detailed information is provided in Chapter IV. After the resource combination for the project is defined in the first phase of the model, the bee-identified schedule is tracked in operation which is the second aspect of the model. This is simply a real-time monitoring of the proposed schedule. If there is an interruption by any means in the schedule, a rescheduling process is performed accordingly. The proposed improved model is executed with a computer code in the following manner. Figure VI.I provides the diagram for bee-based real-time scheduling model.

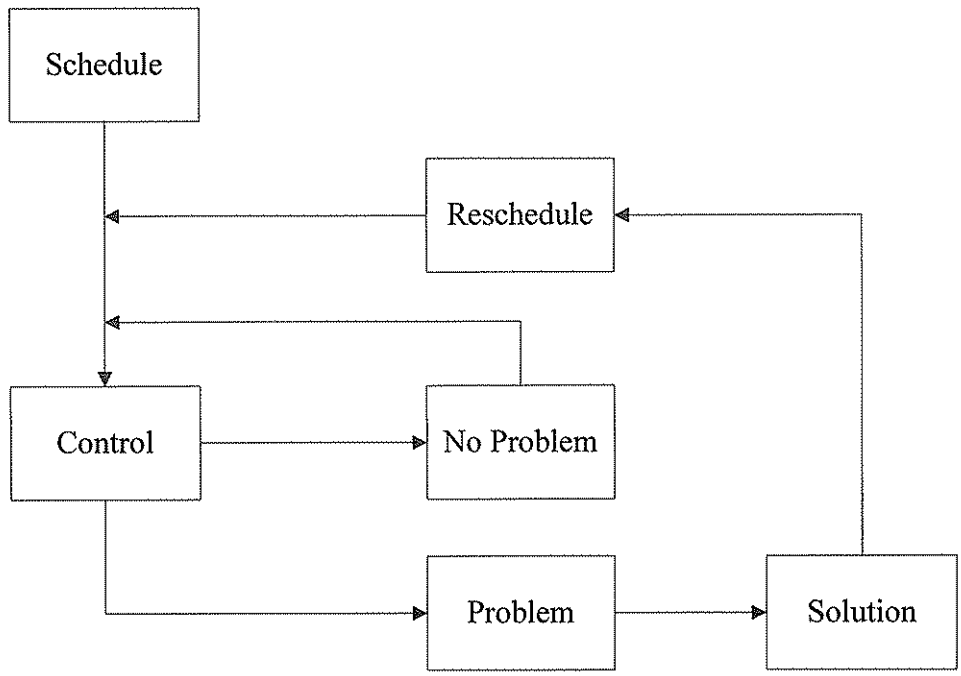

Figure VI.1. Bee-Based Real-Time Scheduling Model

The steps of the model can be summarized as follows:

1. Provide the information about the project.
2. Choose a resource to process each job randomly and compute the completion time for resource activity combination.
3. Repeat the random choice "n" time.
4. Evaluate "n" completion times and select "m" least completion times.
5. Select "e" best solutions among "m" selected sites.
6. Accomplish neighborhood search by changing the resource of jobs on the critical path and calculate completion times. Carry the best solution to the next iteration.
7. Repeat steps from 2 to 6 until the number of iterations is completed.
8. If the solution found is greater than the due date, then calculate the tardy cost. Else go to step 9.
9. Define schedule S for the project giving resource, start and finish times per each activity, and the completion time of the project.
10. Check the availability of resources at each periodic unit time.
11. If all resources are available, go back to step 10 and continue controlling until the end of the project; else go to step 12 and define a new solution.
12. Define the alternatives regarding the unavailability of the resources (temporary or permanent) and add the outsource resources among the alternatives.
13. Go to step 2 and start rescheduling the project.

Step 1. In order to test the model for different scales of projects, a computer code in JAVA language is prepared. The information of the project tested is given to the code with a source file. This file contains the information about the number of jobs of the project, number of resources, resource availability per unit time, resource usage per each job, number of successors and successors for each job. Table VI.1. shows a representation of a source file.

In Table VI.1 the first item in the first line is the number of jobs and the second item is the number of resources. The second line shows the resource availabilities per unit time. The preceding lines provide information for each point of the project. For example, the first two items on third line are the processing time of the activities on each resource. If a job cannot be processed on a resource then the

duration is given as "0". The third item on the third line is the number of successors for that job with the continuing successors. The first and last jobs are the dummy points which show the start and finish points of the project. With the successor information given in the source file the project network is prepared. Taking these into account the following line of the table (5, 1, 2, 17, 19) can be explained. In this line 5 indicates the duration of job 11 on resource 1, and 1 is the duration of job 11 on resource 2. The number on third column (2) is the number of successors, job 11 has two successors and these are jobs 17 and 19 which are the next two numbers on the line.

Table VI.1. An Example of Source File

8	2			
1	1			
0	0	1	2	
9	6	2	3	4
1	9	2	6	7
4	7	2	5	9
3	4	2	6	14
10	2	2	8	13
6	5	2	12	13
3	3	2	10	15
3	3	2	10	14
1	10	2	11	12
5	1	2	17	19
10	7	2	16	18
1	4	1	15	
10	1	2	17	18
6	8	1	16	
9	4	1	21	
1	2	1	20	
6	4	1	19	
5	5	1	20	
6	3	1	21	
5	6	1	22	
0	0	0		

Step 2. A resource is chosen for processing each job randomly. So that, a resource job combination is formed to calculate the project finish time according to

chosen resources. However, the sequence of the jobs to be processed also effects the completion time and it is identified randomly regarding the completion of predecessors. This means that if the predecessors of four jobs are processed and there are 3 jobs to be processed on the same resource in line then the one to be first processed is chosen randomly. The list of each alternative for the sample PSP is given in Appendix 2.

Step 3. The random choice is repeated n times. Project completion times are calculated for each random choice, which is n "number of scout bees" of the Bee System. The scout bees return to the hive with the information of the flower patches in Bee System and in PSP bees carry the information about the completion times. The completion time of a project is the longest path from the starting node to the finishing node of the project. The total completion times of the projects for n alternatives are calculated using the following formula:

$$TC1 = \sum_{i=1}^{k} a \,.\, p_{ij}$$

$$TC2 = \sum_{i=1}^{k} a \,.\, p_{ij}$$

$$\dots$$

$$TCn = \sum_{i=1}^{k} a \,.\, p_{ij}$$

Where;

TC : total completion,

p_{ij} : completion time of job i on resource j (which is chosen randomly for that job),

a : crtitical path factor,

and note that a = 0 if i is not on critical path, else a=1 if i is on critical path.

In the first iteration in sample PSP total completion times are calculated as {27, 28, 24, 26, 23, 24, 26, 29, 20, 22}.

Step 4. The n completion times are evaluated, so that the "m" least completion times among "n" are selected. The "m" is the number of selected sites in Bee System. In sample PSP n is selected 10, and m is 2. The completion times of m solutions are {20, 22}.

Step 5. In the next evaluation "e" best solutions are chosen among the "m" selected sites. These are expected to be more promising sites. For the sample PSP e is determined as 1. The total completion time of e is {20}.

Step 6. After identifying the values of "n", "m", and "e" completion time, the neighbor search begins. For the neighbor search the resource of a job is changed and the completion time is calculated again. The choice of the job on the critical path and the resource is random. More neighborhood search is accomplished around "e" sites and less around "m-e" sites. The best solution found after neighbor search is carried to the next iteration. For the sample problem, 17 time unit is found in the first iteration and carried to the next iteration.

Step 7. The steps from 2 to 6 are repeated until the number of iterations identified for the code is completed. After 10 iterations of the sample PSP, 15 time unit is found.

Step 8. If the solution found is greater than the due date, then a tardy cost is calculated. In sample PSP there is no tardy cost.

Step 9. The schedule S is defined for the project giving resource, start and finish times per each activity, and the completion time of the project. The code prepared for the problem also calculates total cost of resources and the tardy cost if available. In Figure VI.2 a Gantt Chart is provided for the Sample PSP.

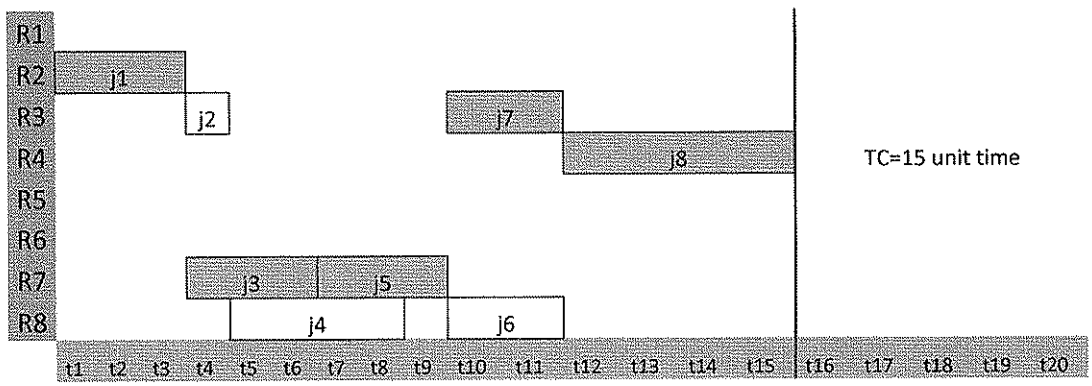

Figure VI.2. Gantt Chart of the Sample Project

Step 10. After defining S, the availability of resources is checked at each periodic unit time. If all resources are available, the system continues controlling; else a new solution for the new situation must be defined.

In order to check if resources are available the following calculation is accomplished; calculate

$$\sum_{j=1}^{m} A_j$$

A_j: Resource availability factor for resource j

If j is available at time t then $A_j = 1$, else $A_j=0$.

Stop when t=TC.

Step 11.a. If $\sum_{j=1}^{l} A_j = 1$, then go back to step 7 and continue with the schedule S.

m: the number of resources available at time t.

Step 11.b. If $\sum_{j=1}^{l} A_j < 1$, and there is an $A_j = 0$, with no job i being processed at j is on time t, then go back to step 7 and continue with the schedule S.

Step 11.c. If $\sum_{j=1}^{l} A_j < 1$, and there is an $A_j = 0$, with job i processed at j is on time t, then go to step 14 to prepare a new schedule S'. For example in the Sample Project a permanent resource crisis occurs on resource R7 from time t=4 to time t=10. In this situation, if the activity on R7 which is j3 waits until the crisis is solved, the project is delayed 6 unit time. This means that the jobs are shifted until the resource becomes available and completion time becomes 15+6=21 time unit. Figure VI.3 shows the Gantt Chart for this shifted situation.

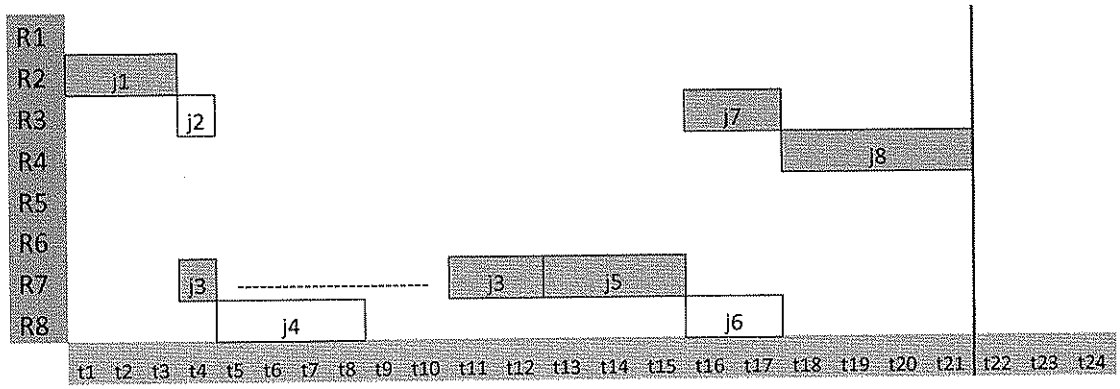

Figure VI.3. Gantt Chart for the Sample Project in Crisis

The other alternative is to reschedule the project with the alternative resources. For rescheduling the project the new resource alternatives are defined in Step 12.

Step 12.a. If the resource is temporarily unavailable and there is an alternative inhouse resource, then prepare the project network using the new resource situation to reschedule and go to step 2. The new situation is defined according to the conditions of the alternative inhouse resource and the condition of the unavailable

resource taking into account the time passes until the unavailable resource becomes available.

Step 12.b. If resource j is temporarily unavailable and there is not an alternative resource, then prepare the project network using an outsource resource situation to reschedule and go to step 2. The new situation is defined according to the conditions of the alternative outsource resource and the condition of the unavailable resource taking into account the time passes until the unavailable resource becomes available.

Step 12.c. If the resource is permanently unavailable and there is an alternative inhouse resource, then prepare the project network using the new resource situation to reschedule and go to step 2. The new situation is defined according to the conditions of the alternative inhouse resource.

Step 12.d. If resource j is permanently unavailable and there is not an alternative resource, then prepare the project network using an outsource resource situation to reschedule and go to step 2. The new situation is defined according to the conditions of the alternative outsource resource.

Step 13. The reschedule of the project is defined after solving the problem with BA. The shortest completion for the sample problem when R7 is unavailable between t=4 to t=10; is 19 time unit. Figure VI.4 provides the reschedule of the shortest completion time calculated by BA under the crisis condition of R7.

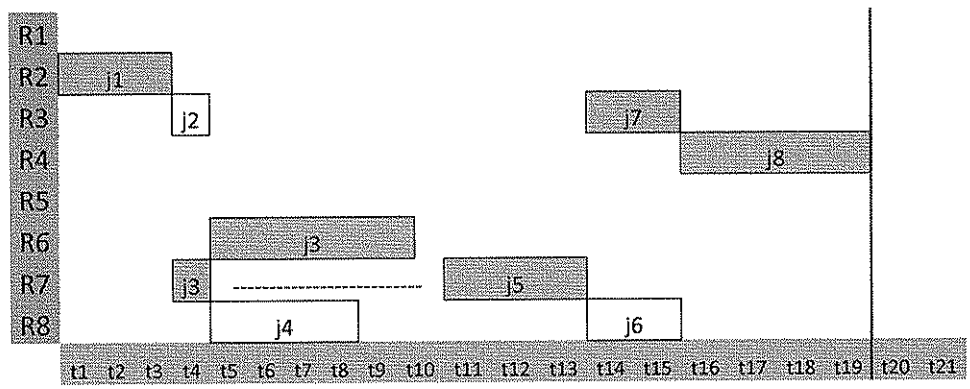

Figure VI.4. Reschedule of the Sample Project with Resource Crisis

CHAPTER VII

IMPLEMENTATION AND EXPERIMENTS

A small scale problem is solved by using the model in Chapter VI however; big scale project problems are needed to be solved for the implementation of the method. For the big scale problems "PROGEN" a project generator program is used. PROGEN generates a general class of precedence and resource constrained project scheduling problems (Kolisch et al., 1992). PROGEN also allows its user to generate instances under controlled conditions. The problems tested in this study are resource-constrained, single-resource, multi-mode PSPs with different scales and they are generated by PROGEN.

For generation PROGEN needs a list of inputs provided. The input file provides information for the properties of the project to be generated, modes, network, resources and limit of iterations. An example of the input file is as follows:

- PROJECTS: In this section number of projects, number of jobs for projects is provided in the input file. For instance:

```
NrOfPro        : 1    & number of projects
MinJob         : 10   & minimal number of jobs per project
MaxJob         : 10   & maximal number of jobs per project
MaxRelDate     : 0    & maximal release date
DueDateFactor : 0.0   & maximal due date
```

- MODES: Number of modes with a range of minimum and maximum duration for the jobs are provided in this section. For instance:

```
MinMode        : 2    & minimal number of modes
MaxMode        : 2    & maximal number of modes
MinDur         : 1    & minimal duration
MaxDur         : 10   & maximal duration
```

- NETWORK: Number of start and finish activities per project and maximum number of successor and predecessor activities per job are identified in this section. For instance:

```
MinOutSource : 1     & minimal number of start activities per project
MaxOutSource : 1     & maximal number of start activities per project
MaxOut       : 3     & maximal number of successor per job
MinInSink    : 1     & minimal number of finish activities per project
MaxInSink    : 1     & maximal number of finish activities per project
MaxIn        : 3     & maximal number of predecessors
Complexity   : 1.5   & complexity of network
```

- RESSOURCEREQUEST/AVAILABILITY: Number of renewable and nonrenewable resources, minimum and maximum demand of these resources and number of resources are provided in this section. For instance:

```
Rmin         : 2     & minimal number of renewable resources
Rmax         : 2     & maximal number of renewable resources
RminDemand   : 1     & minimal (per period) demand
RmaxDemand   : 1     & maximal (per period) demand
RRMin        : 1     & minimal number of resources requested
RRMax        : 1     & maximal number of resources requested
RRF          : 0.5   & resource factor
RRS          : 0.2   & resource strength
Number R-Func. : 2
 p1          : 0.0   & probability to choose a constant function
 p2          : 1.0   & probability yo choose a decreasing function
Nmin         : 0     & cf. renewable resources
Nmax         : 0
NminDemand   : 0
NmaxDemand   : 0
NRMin        : 0
NRMax        : 0
NRF          : 0.0
NRS          : 0.0
```

Number N-Func: 2

p1 : 1.0

p2 : 0.0

Dmin : 0 & cf. renewable resources

Dmax : 0

DminDemand : 0

DmaxDemand : 0

DRMin : 0

DRMax : 0

DRF : 0.0

DRST : 0.0

DRSP : 0.0

Number D-Func.: 2

p1 : 1.0

p2 : 0.0

- LIMIT OF ITERATIONS: maximum numner of trials to generate are given in this section. For instance:

Tolerence Network: 0.05 & tolerated complexity deviation

Tolerence RF : 0.05 & tolerated resouce factor deviation

MaxTrials : 200 & maximal number of trials

With the input file a random number and number of instances are requested by PROGEN. After providing all the information PROGEN generates the project in the size and properties required. The detailed properties of the PROGEN generated projects will be given throughout the text. An example of the project output PROGEN generates is provided in Appendix 3 and its source code which is converted for the computer code is provided in Appendix 4.

The experiments provided in implementation are accomplished on a group of 10-job, 20-job, 30-job, 50-job and 80-job projects with 100, 1 000, and 10 000 iterations. The experiments are separated in two sections. The former experiments find the shortest completion time of the projects with BA and the latter experiments does the same in addition with solving the resource crisis occurred on each scale of a problem.

VII.1. FINDING THE SHORTEST COMPLETION TIME WITH BA

The experiments in this section are tested on 10 different problems with different scales. However; there are some the common point of projects. One of them is the type of resources, which are renewable, nonrenewable resources are not used in these sets of problems. Each resource is available one unit per unit time and a job can only use one resource and one unit of that resource per unit time in each mode. Every project tested is multi-mode but numbers of modes vary depending on the scale. A due date is given for each project but this one is provided under infinite resource constraints. A tardy cost per unit time is also identified for each project.

The parameters of the bee algorithm are chosen the same for each set of problems. These are identified as:

- Number of scout bees (n) are 40.
- Number of selected sites (m) are 20.
- Number of best sites (e) are 110.
- Number of bees recruited for best e sites (nep) are 13.
- Number of bees recruited for the other (m-e) selected sites (nsp) are 3.

The first group of problems is 10-job projects with 2 renewable resources, 2 modes, and a single start and finish activity. Each job can have at most 3 successors and predecessors. 10 projects are tested with 100, 1 000, 10 000 iterations in this group and the solutions are listed in Table VII.1.

Table VII.1. Project completion times of 10-job projects

#jobs	#res	#modes	duedate	10000 it.	1000 it.	100 it.	tardy cost	% deviation
10	2	2	14	**19**	**19**	**19**	**30**	-35,714
10	2	2	16	**18**	**18**	**18**	**8**	-12,500
10	2	2	16	**19**	**19**	**19**	**24**	-18,750
10	2	2	17	**22**	**22**	**22**	**10**	-29,412
10	2	2	19	**20**	**20**	**20**	**9**	-5,263
10	2	2	23	**30**	**30**	**30**	**21**	-30,435
10	2	2	24	**28**	**28**	**28**	**4**	-16,667
10	2	2	25	**30**	**30**	**30**	**5**	-20,000
10	2	2	26	**28**	**28**	**28**	**16**	-7,692
10	2	2	28	**32**	**32**	**32**	**28**	-14,286

As seen in Table VII.1 the solutions reached in three numbers of iterations are the same because the scale of the problem is small enough. There are $2^{10} = 1024$ solutions available.

The second group of problems has 20 jobs where each can be processed on one of the 2 renewable resources. For this group there are $2^{20} = 1\ 048\ 576$ solutions available and the BA find the completion times for the group of 20-job projects. In Table VII.2 solutions of 20-job projects are presented. Unlike the previous group of projects the solutions found in each iteration numbers are different. One of the reasons for this is the scale of the problem. The number of jobs is doubled in this group; however the solution space of the 20-job group is increased to nearly 1 000 times the 10-job ones.

Table VII.2. Project completion times of 20-job projects

#jobs	#res	#modes	duedate	10000 it.	1000 it.	100 it.	tardy cost	% deviation
20	2	2	22	**36**	**36**	37	**112**	-63,636
20	2	2	26	39	**38**	42	**24**	-46,154
20	2	2	33	**40**	**40**	43	**56**	-21,212
20	2	2	38	**42**	44	44	**24**	-10,526
20	2	2	42	**53**	55	57	**165**	-26,190
20	2	2	44	**55**	56	56	**88**	-25,000
20	2	2	45	**57**	59	60	**84**	-26,667
20	2	2	46	**50**	51	54	**64**	-8,696
20	2	2	49	**52**	53	55	**18**	-6,122
20	2	2	53	**57**	59	62	**36**	-7,547

The 30-job projects have also two modes but these are chosen among 4 renewable resources. The size of solution space is $2^{30} = 1\ 073\ 741\ 824$, while the number of jobs are increasing ten by ten, the solution space increases exponentially. The completion times of each number of iterations depart from each other rapidly. In most of the cases 10 000 iterations find the shortest solutions but in some cases less iteration number can find better completion times that is because of the randomness of the system. Even in one case (the project with a due date of 52 unit time) the method finds the best solution in both 1 000 and 10 000 iteration numbers. Table VII.3 provides the project completion times of 30-job projects.

Table VII.3. Project completion times of 30-job projects

#jobs	#res	#modes	duedate	10000 it.	1000 it.	100 it.	tardy cost	% deviation
30	4	2	37	51	**50**	55	**507**	-35,135
30	4	2	40	**45**	52	47	**25**	-12,500
30	4	2	47	**51**	53	53	**76**	-8,511
30	4	2	48	56	**51**	56	**0**	-6,250
30	4	2	48	**53**	**53**	61	**20**	-10,417
30	4	2	52	**52**	**52**	53	**0**	0,000
30	4	2	53	**54**	56	59	**13**	-1,887
30	4	2	58	**60**	63	64	**14**	-3,448
30	4	2	60	**65**	69	68	**0**	-8,333
30	4	2	62	**63**	**63**	64	**25**	-1,613

In the 50-job projects the number of resources increase to 5 renewable ones and number of modes to 3 so that solution space extends to $3^{50} = 7.1789 * 10^{23}$ and this time BA seeks for the shortest completion time among this huge solution space. The Similar like 30-job projects there are some cases where less iteration number finds better completion times. As the number of jobs increase and solution space extends the execution time of the code also increases. Table VII.4 outlines the completion times of 50-job projects.

Table VII.4. Project completion times of 50-job projects

#jobs	#res	#modes	duedate	10000 it.	1000 it.	100 it.	tardy cost	% deviation
50	5	3	32	55	60	**53**	**525**	-65,625
50	5	3	34	**53**	62	64	**399**	-55,882
50	5	3	37	**61**	64	67	**96**	-64,865
50	5	3	38	**58**	68	66	**220**	-52,632
50	5	3	41	**59**	61	61	**666**	-43,902
50	5	3	43	65	**62**	71	**228**	-44,186
50	5	3	44	**60**	61	65	**352**	-36,364
50	5	3	49	**63**	69	70	**140**	-28,571
50	5	3	49	69	**67**	73	**594**	-36,735
50	5	3	50	65	**64**	79	**616**	-28,000

The final group of problems tested in this study is 80-job projects, which also has 5 renewable resources with 3 modes, has at most 5 predecessors and successors. This situation makes the environment more and more complex. The solution space

extends to $3^{80} = 1.478 * 10^{38}$ solutions. Table VII.5 shows the completion times for 80-job projects.

Table VII.5. Project completion times of 80-job projects

#jobs	#res	#modes	duedate	10000 it.	1000 it.	100 it.	tardy cost	% deviation
80	5	3	31	75	76	**72**	**2829**	-132,258
80	5	3	42	**87**	92	93	**2655**	-107,143
80	5	3	42	**96**	98	98	**1620**	-128,571
80	5	3	44	**81**	83	90	**2516**	-84,091
80	5	3	45	**94**	98	95	**1372**	-108,889
80	5	3	53	**96**	105	103	**774**	-81,132
80	5	3	54	89	100	**85**	**806**	-57,407
80	5	3	58	**91**	**91**	96	**957**	-56,897
80	5	3	58	**97**	101	102	**390**	-67,241
80	5	3	59	**91**	92	98	**1632**	-54,237

VII.2. REAL-TIME PROJECT TRACKING WITH BA

For the real-time project tracking experiments a project is chosen for each scale which was solved in previous section and the tracking tests are performed on these. For the chosen projects of 10-job and 20-job, first crisis, and second crisis are applied to the projects respectively. For 30-job, 50-job and 80-job projects a third crisis is added.

For each project, a set of outsources with the completion time and daily cost of each task is also provided for the code. For 10-job and 20-job projects one outsource, for 30-job project two outsources, for the 50-job and 80 job projects three outsources are used. The duration of these outsources are randomly generated integer numbers between 1 and 10; and similarly unit cost of outsources are the same between 2 and 20.

The crisis is identified and given to the code during execution. In addition the cost of resources, due date and unit tardy cost is also provided as the project information in the code. The costs of resources are:

- R1: 7
- R2: 11
- R3: 14
- R4: 16
- R5: 19

After execution, the code calculates the minimum completion time without and with the crisis, resource cost and the tardy cost if it is available. While calculating; the activities executed before the crisis are kept constant and a reschedule is performed for the crisis and remaining activity. For the real-time tracking section two conditions are regarded; the former one uses the in-house resources to solve crisis, and the latter one uses outsources.

Table VII.6 summarizes the results of a 10-job project under the given crisis condition.

Table VII.6. A 10-job project real-time tracking

Project information:						
#jobs: 10						
#resources: 2						
# modes: 2						
due date: 19						
Unit tardy cost: 9						
w/o outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R1	**8**	**10**	**R2**	**12**	**15**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	20		274		9	
1000	20		289		9	
10000	20		274		9	
# iteration	**CT-crisis1**	**Res-cost**	**Trd-cost**	**CT-crisis2**	**Res-cost**	**Trd-cost**
100	20,6	286,2	14,4	22	286,2	27
1000	20,6	286,2	14,4	22	286,2	27
10000	20,6	286,2	14,4	22	286,2	27
with outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R1	**8**	**10**	**R2**	**12**	**15**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	20		274		9	
1000	20		274		9	
10000	20		274		9	
# iteration	**CT-bees**	**Res-cost**	**Trd-cost**	**CT-crisis**	**Res-cost**	**Trd-cost**
100	18	257	0	21	245	18
1000	18	264	0	21	248	18
10000	18	257	0	21	257	18

In Table VII.6 project information is provided in the first section. The second and third sections contain the information about the crisis and the results of the method

employed. For each iteration number the results are summarized. These sections contain:

- the completion time of bees solution: CT-Bees,
- the completion time after solving the crisis: CT-Crisis,
- resource cost: Res-cost,
- tardy cost: Trd-cost.

For the 20-job project, outsources and conditions are similar. The crises are identified for the project. The results are summarized in Table VII.7.

Table VII.7. A 20-job project real-time tracking

Project information:						
#jobs: 20						
#resources: 2						
# modes: 2						
due date: 46						
Unit tardy cost: 16						
w/o outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R2	**17**	**23**	**R1**	**40**	**44**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	54		797		128	
1000	51		733		80	
10000	52		719		96	
# iteration	**CT-crisis1**	**Res-cost**	**Trd-cost**	**CT-crisis2**	**Res-cost**	**Trd-cost**
100	56	772	160	56	772	160
1000	54	725	128	53,5	733,5	120
10000	55	725	144	57	737	176
with outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R2	**17**	**23**	**R1**	**40**	**44**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	52		719		96	
1000	53		773		112	
10000	51		719		80	
# iteration	**CT-bees**	**Res-cost**	**Trd-cost**	**CT-crisis**	**Res-cost**	**Trd-cost**
100	53	683	112	49	654	48
1000	47	640	16	47	640	16
10000	50	657	64	50	655	64

In Table VII.8 the results of the bee based real-time scheduling is provided. In this problem, after the crisis are solved with the usage of outsources it is seen that the project can be completed before the due date.

Table VII.8. A 30-job project real-time tracking

Project information:						
#jobs: 30						
#resources: 4						
# modes: 2						
due date: 52						
Unit tardy cost: 9						
w/o outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R3	**26**	**29**	**R2**	**38**	**42**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	52		1415		0	
1000	53		1324		9	
10000	52		1367		0	
# iteration	**CT-crisis1**	**Res-cost**	**Trd-cost**	**CT-crisis2**	**Res-cost**	**Trd-cost**
100	52	1380	0	53,55	1440,11	13,86
1000	52,3	1347,3	2,61	53,62	1362,71	14,49
10000	52	1326	0	53,11	1372,22	9,9
with outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R3	**26**	**29**	**R2**	**38**	**42**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	56		1203		36	
1000	54		1213		18	
10000	52		1256		0	
# iteration	**CT-bees**	**Res-cost**	**Trd-cost**	**CT-crisis**	**Res-cost**	**Trd-cost**
100	53	1139	9	52	1173	0
1000	53	1247	9	52	1240	0
10000	52	1385	0	51,55	1392	0

Table VII.9 provides the bees and real-time tracking results for a 50-job project. In the computer code in order to use outsources a usage probability is given to the code. For the problems in this section outsource usage probability is identified as 0.5 for conditions outsource is used, else this probability is defined as 0.

Also in the 50-job problem the usage of outsources to solve the crisis, decreases the completion time of the project.

Table VII.9. A 50-job project real-time tracking

Project information:						
#jobs: 50						
#resources: 5						
# modes: 3						
due date: 50						
Unit tardy cost: 44						
w/o outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R5	**32**	**36**	**R3**	**42**	**47**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	67		3000		748	
1000	67		2827		748	
10000	65		2547		660	
# iteration	**CT-crisis1**	**Res-cost**	**Trd-cost**	**CT-crisis2**	**Res-cost**	**Trd-cost**
100	62,33	2872,86	542,08	62,33	2872,86	542,08
1000	65	2735	660	64	2734	616
10000	65	2561	660	64,1	2559,38	619,96
with outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R5	**32**	**36**	**R3**	**42**	**47**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	69		2682		836	
1000	68		2735		792	
10000	61		2687		484	
# iteration	**CT-bees**	**Res-cost**	**Trd-cost**	**CT-crisis**	**Res-cost**	**Trd-cost**
100	67	2602	748	64	2520	616
1000	66	2573	704	62	2550	528
10000	62	2721	528	61,25	2705	495

During the tests it is assumed that the first and last job never have a crisis. If the occurring crisis ends up after the completion time of the project or if a solution is found which completes the project before the end of crisis, the code continues to execute. In this situation the last solution found by the code is regarded as the final solution.

Finally, the last problem solved is an 80-job project. The results are provided in Table VII.10. It is once more seen that usage of outsources decrease the completion time of the project. However, the completion time found by the method is far from catching the due date of the project.

Table VII.10. An 80-job project real-time tracking

Project information:						
#jobs: 80						
#resources: 5						
# modes: 3						
due date: 59						
Unit tardy cost: 51						
w/o outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R1	**24**	**31**	**R4**	**65**	**70**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	115		5318		2856	
1000	114		5731		2805	
10000	107		5373		2448	
# iteration	**CT-crisis1**	**Res-cost**	**Trd-cost**	**CT-crisis2**	**Res-cost**	**Trd-cost**
100	111	5366	2652	110	4996	2601
1000	113	5639	2754	105	5244,6	2346
10000	94	5110	1785	92	4904,66	1683
with outsource	**Resource**	**Start**	**Finish**	**Resource**	**Start**	**Finish**
	R1	**24**	**31**	**R4**	**65**	**70**
# iteration	**CT-bees**		**Res-cost**		**Trd-cost**	
100	106		5232		2397	
1000	111		5625		2652	
10000	102		5479		2193	
# iteration	**CT-bees**	**Res-cost**	**Trd-cost**	**CT-crisis**	**Res-cost**	**Trd-cost**
100	103	5286,2	2244	101	5076,93	2142
1000	99	5236	2040	89	4855	1530
10000	102	5039	2193	95	4926	1836

CHAPTER VIII

CONCLUSIONS AND FUTUREWORKS

The possibility of finding a shortest completion time for the single-resource, multi-mode, resource-constrained PSP with BCO is proven. The first section of the model is designed to distinguish the best solution among the alternatively available ones. In order to solve the problem as a basic model the resource availability is kept a minimum and is taken the value of 1 unit per unit time. To ensure that limitation a project generator "PROGEN" is used for the test problems. The problems are generated for the specified conditions and tests are successfully performed on these. In the former case, a small scale (8-job) project is solved and explained in detail for the proof of concept. In the latter cases, the method is tested for sets of different scale problems ranging from 10-job to 80-job. The changing number of modes has increased the size of the solution space exponentially. Despite the huge size, the method scheduled projects and calculated the shortest completion times, rapidly.

The capability of BCO enabling the solution of real-time scheduling problems and generating respective dynamic schedules for even extremely complex problems are also verified. The model is tested on some of the tests problems in the case of resource crisis for a given duration. The model has the ability to keep the schedule constant up to the crisis and generate a reschedule for the remaining jobs after the crisis. In this aspect the model is thought to have the ability to track the project in real-time and decide among the new conditions defined by the scheduler if a crisis occurs.

In this thesis a model is designed to find the shortest completion time to a single-resource, multi-mode, resource-constrained PSP problem and with tracking the changes of resource environment in real-time, adopting the project to the new conditions. However, some adjustments and improvements can be accomplished for further extensions of the research:

- finding the shortest completion time for the single-resource, multi-mode, resource-constrained PSP problem with multi-resource availability,
- finding the shortest completion time for the multi-resource multi-mode, resource-constrained PSP problem,
- finding the shortest completion time for the single-resource, multi-mode, resource-constrained multi-project case,
- finding the shortest completion time for the multi-resource multi-mode, resource-constrained multi-project case,
- multi-objective PSP problem, and
- comparison with the other methods.

REFERENCES

[1] California Technology Agency; "Project Management Planning", *Project Management Methodology*, January **(1997)** 1-7. http://www.cio.ca.gov/Government/IT_Policy/pdf/PM3.4_Planning_Project_Schedule.pdf (07.01.2009).

[2] Chong, C.S.; Low, M.Y.H.; Sivakumar, A.I.; Gay, K.L.: "A Bee Colony Optimization Algorithm to Job Shop Scheduling", *Proceedings of the 2006 Winter Simulation Conference*, Perrone, L.F.; Wieland, F.P.; Liu, J.; Lawson, B.G.; Nicol, D.M.; Fujimoto, R.M.Editors; **(2006)** 1954-1961.

[3] Cox, M.D.; Myerscough, M.R.: "A Flexible Model of Foraging by a Honey Bee Colony: The Effects of Individual Behaviour on Foraging Success", *Journal of Theoretical Biology*, Elsevier, 223, **(2003)** 179-197.

[4] Doucet, J.: "Project Management,: Challenges and Lessons Learned", *Bureau of Economic Jeology 663*, 09 February 2007, http://www.beg.utexas.edu/energyecon/ua_2007/AB_Project_Mnt_challenges.pdf (01.07.2009)

[5] Egeland, B.; "Project Managers Top Ten Challenges", **(2007)**, http://pmtips.net/project-managers-top-ten-challenges/ (01.07.2009).

[6] Ford, P.W.; "Top 10 Project Management Challenges", **(2004)**, http://www.projectmanagementcourse.com/project-challenges.html (01.07.2009)

[7] Gray, C.F.; Larson, E.W.: *Project Management: The Managerial Process*, 4th Edition, McGraw Hill Companies Inc., New York, USA, **(2008)**.

[8] Herroelen, W.; Leus, R.: "Project Scheduling Under Uncertainty: Survey and Research Potentials", *European Journal of Operational Research*, Elsevier, 165, 2, **(2005)** 289-306.

[9] Karaboğa, D.: "An Idea Based on Honey Bee Swarm for Numerical Optimization", Technical Report – TR 06, Erciyes University, Engineering Faculty, Computer Engineering Department, **(2005)**, http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf (01.07.2009).

[10] Karaboğa, D.; Basturk,B.: "A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", *Journal of Global Optimization*, 39, 3, **(2007)** 459-171.

[11] Karaboğa, D.; Akay, B.: "Proportional-Integral-Derivative Controller Design by Using Artificial Bee Colony, Harmony Search, and the Bees Algorithms", *Journal of Systems and Control Engineering*, Proc. IMechE, Vol. 224 Part I, **(2010)**, 869-883.

[12] Karaboğa, D.; Akay, B.: "Solving Large Scale Numerical Problems using Artificial Bee Colony Algorithm", *Proceedings of 6th International symposium on Intelligent and Manufacturing Systems*, Sakarya, Turkey, October 14-16, **(2008)** 401-410.

[13] Knotts, G.; Dror, M.; Hartmann, B.C.; "Agent-Based Project Scheduling", *IIE Transactions*, 32, **(2000)**, 387-401, http://www.cs.bgu.ac.il/~discsp/papers/Proj_Agnt_Sched.pdf (07.01.2009).

[14] Kolisch, R.; Hartmann, S.: "Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An update", *European Journal of Operational Research*, Elsevier, 174 **(2006)** 23-27.

[15] Kolisch, R.; Padman, R.: "An Integrated Survey of Deterministic Project Scheduling", *The International Journal of Management Science*, Pergamon, 29 **(2001)** 249-272.

[16] Kolisch, R.; Sprecher, A.; Drexl, A.: "Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problem", *Manuskripte Aus Den Instituten Für Betriebswirtschaftslehre Der Universitat Kiel*, Nr. 301, December, (**1992**) 1-35.

[17] Lanchester, J.; Özbayrak, M.: "Evolutionary Algorithms Applied to Project Scheduling Problems - A Survey of the State - of - Art", *International Journal of Production Research*, Taylor and Francis, Vol. 45, No. 2, Jan 15, **(2007)**, 425-450.

[18] Lemmens, N.; de Jong, S.; Tuyls, K.; Nowe, A.: "A bee algorithm for multi-agent systems", AAMAS, Honolulu, Hawai, USA, **(2007)**.

[19] Lu, X.; Zhou, Y.: "A Novel Global Convergence Algorithm: Bee Collecting Pollen Algorithm", *Proceedings of 4th International Conference on Intelligent Computing*, ICIC 2008 Shanghai, China, September 15-18, **(2008)** 518-525.

[20] Mohammadi, A.; Akl, S.G.: "Scheduling Algorithms for Real-Time Systems", *Technical Report No.: 2005 – 499*, School of Computing, Queen's University, Kingston, Ontario, Canada, **(2005)**.

[21] Nakrani, S.; Tovey, C.: "On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers", *Adaptive Behaviour*, **(2004)**, 12(3-4):pp.223-240.

[22] Öztürk, C.; Karaboğa, D.: "Classification by Neural Networks and Clustering with Artificial Bee Colony (ABC) Algorithm", *Proceedings of 6th International symposium on Intelligent and Manufacturing Systems*, Sakarya, Turkey, October 14-16, **(2008)** 411-419.

[23] Pham, D.T.; Afify, A.A.; Koç, E.: "Manufacturing Cell Formation Using the Bees Algorithm", *The IPROMS Innovative Production Machines and Systems Conference*, Cardiff, Wales, United Kingdom, July, **(2007a)**.

[24] Pham, D.T.; Al-Jabbouli, H.; Mahmuddin, Otri, S.; Haj Darwish, A.: "Application of the Bees Algorithm to Fuzzy Clustering", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008a)**.

[25] Pham, D.T.; Ang, M.C.; Ng, K.W; Otri, S.; Haj Darwish, A.: "Generating Branded Product Concepts: Comparing the Bees Algorithm and an Evolutionary Algorithm", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008b)**.

[26] Pham, D.T.; Castellani, M.; Fahmy, A.A.: "Learning the Inverse Kinematics of a Robot Manipulator Using the Bees Algorithm", *The IEEE International Conference on Industrial Informatics*, DCC, Daejon, Korea, July 13-16, **(2008c)** 493-498.

[27] Pham, D.T.; Ghanbarzadeh, A.; "Multi – Objective Optimization Using the Bees Algorithm", *The IPROMS Innovative Production Machines and Systems Conference*, Cardiff, Wahles, United Kingdom, July, **(2007b)**.

[28] Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Rahim, S.; Zaidi, M.: "The bees algorithm – A Novel Tool for Complex Optimization Problems", *Intelligent Production Machines and Systems*, Pham, D.T.; Eldukhri, E.E.; Soroka, A.J. Editors.; Elsevier, **(2006a)** 454-459.

[29] Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Rahim, S.; Zaidi, M.: "Bee Algorithm A Novel Approach to Function Optimization", Technical Note: MEC 0501, The Manufacturing Engineering Center, Cardiff University, **(2005)** 1-40.

[30] Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.: "Application of the Bees Algorithm to the Training of Radial Basis Function Networks for Control Chart Pattern Recognition", *Proceedings of the 5th International Seminar on Intelligent Computation in Manufacturing Engineering*, CIRP ICME'06, Ischia, Italy, **(2006b)**.

[31] Pham, D.T.; Haj Darwish, A.: "Fuzzy Selection of Local Search Sites in the Bees Algorithm", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008)**.

[32] Pham, D.T.; Haj Darwish, A.: "Using the Bees Algorithm with Kalman Filtering to Train An Artificial Neural Network for Pattern Classification", *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 224, No. 7, **(2010)**, 885-892.

[33] Pham, D.T.; Haj Darwish, A.; Eldukhri, E.E.; Otri, S.: "Using the Bees Algorithm to Tune a Fuzzy Logic Controller for a Robot Gymnast", *The IPROMS Innovative Production Machines and Systems Conference*, Cardiff, Wales, United Kingdom, July, **(2007c)**.

[34] Pham, D.T.; Koç, E.: "Flowshop Sequencing Using the Bees Algorithm", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008)**.

[35] Pham, D.T.; Koç, E.; Ghanbarzadeh, A.; Otri, S.: "Optimization of the Weights of Multi-Layered Perceptrons Using the Bees Algorithm", *Proceedings of the 5th International Symposium on Intelligent Manufacturing Systems*, May 29-31, **(2006c)** 38-46.

[36] Pham, D.T.; Koç, E.; Kalyoncu, M.; Tınkır, M.: "Hierarchical PID Controller Design for a Flexible Link Robot Manipulator Using the Bees Algorithm", *6th International Symposium on Intelligent an Manufacturing Systems*, Sakarya, Türkiye, October, **(2008d)**.

[37] Pham, D.T.; Koç, E.; Lee, J.Y.; Phrueksanant, J.: "Using the Bees Algorithm to Schedule Jobs for a Machine", *Proceedings of the 8th International Conference*

on Laser Methodology, CMM and Machine Tool Performance, Euspen, Cardiff, UK, **(2007d)** 430-439.

[38] Pham, D.T.; Lee, J.Y.; Darwish, A.H.: "Multi-Objective Environmental / Economic Dispatch Using the Bees Algorithm with Pareto Optimality and Weighted Sum", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008e)**.

[39] Pham, D.T.; Muhamad, Z.; Mahmuddin, M.; Ghanbarzadeh, A.; Koç, E.; Otri, S.: "Using the Bees Algorithm to Optimise a Support Vector Machine for Wood Defect Classification", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2007e)**.

[40] Pham, D.T.; Negm, M.A.; ; Otri, S.: "Using the Bees Algorithm to Solve a Stochastic Optimization Problem", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008f)**.

[41] Pham, D.T.; Otri, S.; Afify, A.; Mahmuddin, M.; Al-Jabbouli, H.: "Data Clustering Using the Bees Algorithm", *Proceedings of 40th CIRP Int. Manufacturing Systems Seminar*, Liverpool, United Kingdom, **(2007f)**.

[42] Pham, D.T.; Otri, S.; Ghanbarzadeh, A.; Koç, E.: "Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition", *ICCTA*, Syria, **(2006d)** 1624-1629.

[43] Pham, D.T.; Sholedolu, M.: "Using a Hybrid PSO-Bees Algorithm to Train Neural Networks for Wood Defect Classification", *The IPROMS Innovative Production Machines and Systems Virtual Conference*, Cardiff, Wales, United Kingdom, July, **(2008)**.

[44] Pham, D.T.; Soroka, A.J.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Packianather, M.: "Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm", *Proceedings of the International Conference on Industrial Informatics*, Singapore, **(2006e)**.

[45] Pinedo, M.: *Scheduling: Theory, Algorithms, and Systems*, 2nd Edition, Prentice Hall Inc., New Jersey, USA, **(2002)**.

[46] Quijana, N.; Passino, K.M.: "Honey Bee Social Foraging Algorithms for Resource Allocation, Part I: Algorithm and Theory", *Proceedings of the 2007 American Control Conference*, New York City, USA. July 11-13, **(2007a)**.

[47] Quijana, N.; Passino, K.M.: "Honey Bee Social Foraging Algorithms for Resource Allocation, Part II: Application", *Proceedings of the 2007 American Control Conference*, New York City, USA. July 11-13; **(2007b)**.

[48] Sato, T.; Hagiwara, M.: "Bee system: Finding Solution by A Concentrated Search", *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics 'Computational Cybernetics and Simulation*, Orlando, FL, USA, **(1997)** 3954–3959.

[49] Sha, L.; Abdelzaher, T.; Arzen, K.-E.; Cervin, A.; Baker, T.; Burns, A.; Buttazzo, G.; Caccamo, M.; Lehoczky, J.; Mok, A.K.: "Real Time Scheduling Theory: A Historical Perspective", *Real Time Systems*, 28, 101-155, **(2004)**.

[50] Shumsky, R.A.: "A note on Project Scheduling", William E. Simon Graduate School of Business Administration, University of Rochester, (2003), 1-10, http://mba.tuck.dartmouth.edu/pages/faculty/robert.shumsky/Project_scheduling_note.pdf (07.01.2009).

[51] Teodorovic, D.; Dell' Orco, M.: "Mitigating Traffic Congestion: Solving the Ride – Matching Problem by Bee Colony Optimization", *Transportation Planning and Technology*, Routledge, Vol. 31, No. 2, April **(2008)**, 135-152.

[52] Teodorovic, D.; Lucic, P.; Markovic, G.; Dell' Orco, M.: "Bee Colony Optimization: Principles and Applications", *8th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, Serbia, September 25-27, **(2006)** 151-156.

[53] Wong, L.; Low, M.Y.H.; Chong, C.S.: "Bee Colony Optimization for Traveling Salesman Problem", *Second Asia International Conference on Modelling and Simulation*, Kuala Lumpur, Malaysia, May 13-15, **(2008a)** 818-823.

[54] Wong, L.; Low, M.Y.H.; Chong, C.S.: "Bee Colony Optimization with Local Search for Traveling Salesman Problem", *The IEEE International Conference on Industrial Informatics*, DCC, Daejon, Korea, July 13-16, **(2008b)** 1019-1025.

[55] Wong, L.; Puan, C.Y.; Low, M.Y.H.; Chong, C.S.: "Bee Colony Optimization Algorithm with Big Valley Landscape Exploitation for Job Shop Scheduling Problems", Proceedings of the 2008 Winter Simulation Conference, **(2008c)** 2050-2058.

[56] Yonezawa, Y.; Kikuchi, T.: "Ecological Algorithm for Optimal Ordering Used by Collective Honey Bee Behavior", *7th International Symposium on Micro Machine and Human Science*, **(1996)** 249-256.

APPENDIX

Appendix 1 - Research on BCO

Author(s) & Year	Methodology	Research Area	Major Findings
Yonezawa & Kikuchi (1996)	Theory Building	Collective honey bee behavior	An algorithm being generated with the co-operative behavior of bees, being effective in speed of decision and solution of complex problems.
Sato & Hagiwara (1997)	Theory Building	Improving GA with BS	Improved GA is being obtained high ability for local search without degrading the global search ability, validity of the BS and BS being more effective especially for highly complex multivariate functions.
Cox & Myerscough (2003)	Theory Building	A model of foraging by a honey bee colony	A heterogeneous colony may perform better than a homogeneous colony in foraging behavior.
Chong et al. (2006)	Case Study / Comparative Study	BCO algorithm to job shop scheduling	The performance of BCO algorithm being comparable to ACO algorithms but gapping behind the efficient TS heuristics.
Pham et al. (2006a)	Theory Building	BA	BA is being introduced to literature and having remarkable robustness with 100% success in all cases of multi-modal functions in n-dimensions.
Pham et al. (2006b)	Case Study / Comparative Study	Application of BA to the training of RBF networks for the recognition of control chart patterns	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers, BA producing RBF networks with a lower error value than that obtained for an RBF network trained conventionally.
Pham et al. (2006c)	Case Study / Comparative Study	Optimization of weights of MLP using BA	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers than the backpropagation algorithm.
Pham et al. (2006d)	Case Study / Comparative Study	Application of BA to the training of learning vector quantization networks for the recognition of control chart patterns	Despite the high dimensionality of the problem, the BA being still successful to train more accurate classifiers than the standard learning vector quantization training algorithm.
Pham et al. (2006e)	Case Study / Comparative Study	Optimization of neural networks for wood defect detection using BA	The accuracy obtained by the BA being comparable to backpropagation algorithm, and BA being faster.
Teodorovic et al. (2006)	Theory Building / Case Study	BCO Principles and applications	BCO is being proposed; two algorithms being described: BS and FBS; FBS is being capable of solving deterministic combinatorial problems.
Karaboğa & Baştürk (2007)	Theory Building / Comparative Study	ABC Algorithm	ABC is being based on the intelligent behavior of honey bee swarm, and used for optimizing multivariable functions, ABC outperforming GA, PSO, and PSIEA.
Lemmens et al. (2007)	Comparative Study	Comparison of the non-pheromone based algorithm with a pheromone based algorithm in the domain of foraging	Non-pheromone based algorithms being more efficient when finding and collecting food with three times less iterations, less total computation time but being less adaptive than a pheromone based algorithm.

Appendix 1 - Research on BCO (continues...)

Author(s) & Year	Methodology	Research Area	Major Findings
Lemmens et al. (2007)	Comparative Study	Comparison of the non-pheromone based algorithm with a pheromone based algorithm in the domain of foraging	Non-pheromone based algorithms being more efficient when finding and collecting food with three times less iterations, less total computation time but being less adaptive than a pheromone based algorithm.
Pham et al. (2007a)	Case Study	MCF with BA	BA being very effective in generating optimal solutions to the MCF problem.
Pham et al. (2007b)	Case Study	Multi-Objective optimization with BA	A modified version of the BA, BA being computationally fast multi- objective optimizer tool.
Pham et al. (2007c)	Case Study	Tuning a fuzzy logic controller for a robot gymnast using BA	Controller tuned with BA giving a smooth performance with fewer variations and smaller input control signals than in the case of the unturned controller, BA being a useful tool for tuning fuzzy logic controllers to achieve better performance.
Pham et al. (2007d)	Case Study	Scheduling jobs for a machine using BA	Being first to report application of BA to a combinatorial optimization problem, performing more strongly than existing techniques, BA being more stable and robust than other algorithms.
Pham et al. (2007e)	Case Study	Optimization of a SVM for wood defect classifier using BA, best combination of SVM parameters and data features to maximize defect classification accuracy	The accuracies obtained being higher than that of obtained by a standard SVM, reconfirming usefulness of BA as an optimization tool.
Pham et al. (2007f)	Case Study / Method Building	Data Clustering using BA	Proposing a clustering method that integrates the simplicity of the k-means algorithm with the capability of BA to avoid local optima, the proposed method not being trapped at locally optimal solutions, and producing better performances than those of the k-means algorithm and the GA-Clustering method.
Quijano & Passino (2007a)	Theory Building	Honey bee social foraging algorithms (Part I)	Ideal free distribution concept from theoretical ecology is being introduced, and ideal free distribution being a global optimum point for both a single hive and multiple hives.
Quijano & Passino (2007b)	Case Study	Honey bee social foraging algorithms (Part II)	Engineering application on honey bee social foraging algorithm.
Karaboğa & Akay (2008)	Theory Building / Comparative Study	Large scale numerical problems with ABC Algorithm	The performance of ABC algorithm being compared to Differential Evolution and PSO for large scale numerical problems, and ABC algorithm having superior performance over Differential Evolution and PSO.
Lu & Zhou (2008)	Theory Building	Bee Collecting Pollen Algorithm	Performance of algorithm being comparable to ACO, but gapping behind the efficient TS heuristics.

Appendix 1 - Research on BCO (continues...)

Author(s) & Year	Methodology	Research Area	Major Findings
Öztürk & Karaboğa (2008)	Case Study	Classification by neural networks and clustering with ABC Algorithm	The performance of ABC algorithm being tested on training of Artificial Neural Networks and clustering for classification of different data sets of cancer, diabetes and heart where ABC algorithm being successfully able to be applied.
Pham et al . (2008a)	Case Study / Comparative Study	Fuzzy Clustering with BA	BA combination with the Fuzzy C-Means Algorithm improving its clustering results significantly and giving better results than the Fuzzy C-Means Algorithm combined with GA.
Pham et al. (2008b)	Case Study / Comparative Study	Generating product design concepts using BA	Comparison of BA with Evolutionary Algorithm for the task of maintaining the brand identity of the product, an application on a new Coca Cola Bottle design, and BA outperforming the Evolutionary Algorithm.
Pham et al. (2008c)	Case Study / Comparative Study	Learning the inverse kinematics of a robot manipulator using BA, training MLP neural networks to model the inverse kinematics of n articulated robot manipulator arm	BA producing accurate learning results for the three modeling tasks, high reliability of BA, superior performance of BA over Backpropagation Algorithm and Evolutionary Algorithm.
Pham & Haj Darwish (2008)	Method Building	Fuzzy Selection of Local Search Sites in BA	A combination of BA as a core and a fuzzy system for the greedy selection is being applied for function optimization, and improved BA being robust and reliable for function optimization.
Pham & Koç (2008)	Case Study	Flowshop Sequencing with BA	BA outperforming other techniques in terms of the efficiency and robustness.
Pham et al. (2008d)	Case Study	Design of a PID controller for a flexible single-link robot manipulator using BA	PID controller which is tuned with BA to minimize vibrations and positioning errors being both efficient and robust.
Pham et al. (2008e)	Case Study / Comparative Study	Multi-objective Optimization of Environmental/Economic Power Dispatch with BA	Minimization of fuel cost and nitrogen oxide emissions simultaneously using BA with Pareto optimality, comparison with other methods, BA being suitable for the problem.
Pham et al. (2008f)	Case Study	Solving A Stochastic Optimization Problem with BA	BA is being used to determine the parameter setting for an ACS to find the shortest route of a TSP, and parameter setting produced by BA outperforming other methods.
Pham & Sholedolu (2008)	Method Building	Combining BA with PSO to train Neural Networks for Wood Defect Classifier	Despite the high dimensionality of the problem, the PSO-BA trained classifiers being able to identify defects more accurately than original PSO algorithm and back propagation method.
Wong et al. (2008a)	Model Building / Comparative Study	BCO for Traveling Salesman Problem (TSP)	A BCO based model being introduced to solve TSP, and model being tested on a set of benchmark problems.

Appendix 1- Research on BCO (continues…)

Author(s) & Year	Methodology	Research Area	Major Findings
Wong et al. (2008b)	Model Building / Comparative Study	BCO with Local Search for TSP	BCO being integrated with 2-opt heuristic to solve TSP, integration improving BCO performance significantly.
Wong et al. (2008c)	Comparative Study	BCO Algorithm with Big Valley Exploitation	Bee Colony Big Valley is being able to achieve comparable performance to TS Algorithm, Shifting Bottleneck Heuristic and BCO Algorithm with neighborhood search using small number of iterations with better performance of computation time.
Davidovic et al. (2010)	Case Study / Comparative Study	BCO for the p-center problem	A variant of BCO based on the improvement concept (BCOi) is proposed, BCOi being a very useful concept for solving difficult combinatorial problems, and generating high-quality solutions.
Karaboga & Akay (2010)	Case Study / Comparative Study	The effect of problem dimensionality on the performance of basic ABC, HS and BA	ABC being showed a slightly better performance on time-delayed and oscillatory PID control systems.
Pham & Darwish (2010)	Methodology / Case Study	BA with Kalman Filtering to train artificial neural network for pattern classification	Combining random exploration and greedy exploitation, giving better test results, reducing number of iterations and speeding up the training manifold.

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
1	R1	5	R3	1	R6	8	R4	9	R2	5	R1	7	R2	4	R6	6	17	13	10	20	17	**31**
2	R2	3	R3	1	R6	8	R4	9	R2	5	R1	7	R2	4	R6	6	17	13	10	20	17	**29**
3	R1	5	R5	6	R6	8	R4	9	R2	5	R1	7	R2	4	R6	6	22	18	15	20	17	**33**
4	R2	3	R5	6	R6	8	R4	9	R2	5	R1	7	R2	4	R6	6	22	18	15	20	17	**31**
5	R1	5	R3	1	R7	3	R4	9	R2	5	R1	7	R2	4	R6	6	17	13	10	15	12	**28**
6	R2	3	R3	1	R7	3	R4	9	R2	5	R1	7	R2	4	R6	6	17	13	10	15	12	**26**
7	R1	5	R5	6	R7	3	R4	9	R2	5	R1	7	R2	4	R6	6	22	18	15	15	12	**33**
8	R2	3	R5	6	R7	3	R4	9	R2	5	R1	7	R2	4	R6	6	22	18	15	15	12	**31**
9	R1	5	R3	1	R6	8	R8	4	R2	5	R1	7	R2	4	R6	6	12	13	10	20	17	**31**
10	R2	3	R3	1	R6	8	R8	4	R2	5	R1	7	R2	4	R6	6	12	13	10	20	17	**29**
11	R1	5	R5	6	R6	8	R8	4	R2	5	R1	7	R2	4	R6	6	17	18	15	20	17	**31**
12	R2	3	R5	6	R6	8	R8	4	R2	5	R1	7	R2	4	R6	6	17	18	15	20	17	**29**
13	R1	5	R3	1	R7	3	R8	4	R2	5	R1	7	R2	4	R6	6	12	13	10	15	12	**26**
14	R2	3	R3	1	R7	3	R8	4	R2	5	R1	7	R2	4	R6	6	12	13	10	15	12	**24**
15	R1	5	R5	6	R7	3	R8	4	R2	5	R1	7	R2	4	R6	6	17	18	15	15	12	**29**
16	R2	3	R5	6	R7	3	R8	4	R2	5	R1	7	R2	4	R6	6	17	18	15	15	12	**27**
17	R1	5	R3	1	R6	8	R4	9	R7	3	R1	7	R2	4	R6	6	17	11	8	18	15	**29**
18	R2	3	R3	1	R6	8	R4	9	R7	3	R1	7	R2	4	R6	6	17	11	8	18	15	**27**
19	R1	5	R5	6	R6	8	R4	9	R7	3	R1	7	R2	4	R6	6	22	16	13	18	15	**33**
20	R2	3	R5	6	R6	8	R4	9	R7	3	R1	7	R2	4	R6	6	22	16	13	18	15	**31**
21	R1	5	R3	1	R7	3	R4	9	R7	3	R1	7	R2	4	R6	6	17	11	8	13	10	**28**
22	R2	3	R3	1	R7	3	R4	9	R7	3	R1	7	R2	4	R6	6	17	11	8	13	10	**26**
23	R1	5	R5	6	R7	3	R4	9	R7	3	R1	7	R2	4	R6	6	22	16	13	13	10	**33**
24	R2	3	R5	6	R7	3	R4	9	R7	3	R1	7	R2	4	R6	6	22	16	13	13	10	**31**
25	R1	5	R3	1	R6	8	R8	4	R7	3	R1	7	R2	4	R6	6	12	11	8	18	15	**29**
26	R2	3	R3	1	R6	8	R8	4	R7	3	R1	7	R2	4	R6	6	12	11	8	18	15	**27**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
27	R1	5	R5	6	R6	8	R8	4	R7	3	R1	7	R2	4	R6	6	17	16	13	18	15	**29**
28	R2	3	R5	6	R6	8	R8	4	R7	3	R1	7	R2	4	R6	6	17	16	13	18	15	**27**
29	R1	5	R3	1	R7	3	R8	4	R7	3	R1	7	R2	4	R6	6	12	11	8	13	10	**24**
30	R2	3	R3	1	R7	3	R8	4	R7	3	R1	7	R2	4	R6	6	12	11	8	13	10	**22**
31	R1	5	R5	6	R7	3	R8	4	R7	3	R1	7	R2	4	R6	6	17	16	13	13	10	**28**
32	R2	3	R5	6	R7	3	R8	4	R7	3	R1	7	R2	4	R6	6	17	16	13	13	10	**26**
33	R1	5	R3	1	R6	8	R4	9	R2	5	R8	2	R2	4	R6	6	12	8	10	15	17	**28**
34	R2	3	R3	1	R6	8	R4	9	R2	5	R8	2	R2	4	R6	6	12	8	10	15	17	**26**
35	R1	5	R5	6	R6	8	R4	9	R2	5	R8	2	R2	4	R6	6	17	13	15	15	17	**28**
36	R2	3	R5	6	R6	8	R4	9	R2	5	R8	2	R2	4	R6	6	17	13	15	15	17	**26**
37	R1	5	R3	1	R7	3	R4	9	R2	5	R8	2	R2	4	R6	6	12	8	10	10	12	**23**
38	R2	3	R3	1	R7	3	R4	9	R2	5	R8	2	R2	4	R6	6	12	8	10	10	12	**21**
39	R1	5	R5	6	R7	3	R4	9	R2	5	R8	2	R2	4	R6	6	17	13	15	10	12	**28**
40	R2	3	R5	6	R7	3	R4	9	R2	5	R8	2	R2	4	R6	6	17	13	15	10	12	**26**
41	R1	5	R3	1	R6	8	R8	4	R2	5	R8	2	R2	4	R6	6	7	8	10	15	17	**28**
42	R2	3	R3	1	R6	8	R8	4	R2	5	R8	2	R2	4	R6	6	7	8	10	15	17	**26**
43	R1	5	R5	6	R6	8	R8	4	R2	5	R8	2	R2	4	R6	6	12	13	15	15	17	**28**
44	R2	3	R5	6	R6	8	R8	4	R2	5	R8	2	R2	4	R6	6	12	13	15	15	17	**26**
45	R1	5	R3	1	R7	3	R8	4	R2	5	R8	2	R2	4	R6	6	7	8	10	10	12	**23**
46	R2	3	R3	1	R7	3	R8	4	R2	5	R8	2	R2	4	R6	6	7	8	10	10	12	**21**
47	R1	5	R5	6	R7	3	R8	4	R2	5	R8	2	R2	4	R6	6	12	13	15	10	12	**26**
48	R2	3	R5	6	R7	3	R8	4	R2	5	R8	2	R2	4	R6	6	12	13	15	10	12	**24**
49	R1	5	R3	1	R6	8	R4	9	R7	3	R8	2	R2	4	R6	6	12	6	8	13	15	**26**
50	R2	3	R3	1	R6	8	R4	9	R7	3	R8	2	R2	4	R6	6	12	6	8	13	15	**24**
51	R1	5	R5	6	R6	8	R4	9	R7	3	R8	2	R2	4	R6	6	17	11	13	13	15	**28**
52	R2	3	R5	6	R6	8	R4	9	R7	3	R8	2	R2	4	R6	6	17	11	13	13	15	**26**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
53	R1	5	R3	1	R7	3	R4	9	R7	3	R8	2	R2	4	R6	6	12	6	8	8	10	**23**
54	R2	3	R3	1	R7	3	R4	9	R7	3	R8	2	R2	4	R6	6	12	6	8	8	10	**21**
55	R1	5	R5	6	R7	3	R4	9	R7	3	R8	2	R2	4	R6	6	17	11	13	8	10	**28**
56	R2	3	R5	6	R7	3	R4	9	R7	3	R8	2	R2	4	R6	6	17	11	13	8	10	**26**
57	R1	5	R3	1	R6	8	R8	4	R7	3	R8	2	R2	4	R6	6	7	6	8	13	15	**26**
58	R2	3	R3	1	R6	8	R8	4	R7	3	R8	2	R2	4	R6	6	7	6	8	13	15	**24**
59	R1	5	R5	6	R6	8	R8	4	R7	3	R8	2	R2	4	R6	6	12	11	13	13	15	**26**
60	R2	3	R5	6	R6	8	R8	4	R7	3	R8	2	R2	4	R6	6	12	11	13	13	15	**24**
61	R1	5	R3	1	R7	3	R8	4	R7	3	R8	2	R2	4	R6	6	7	6	8	8	10	**21**
62	R2	3	R3	1	R7	3	R8	4	R7	3	R8	2	R2	4	R6	6	7	6	8	8	10	**19**
63	R1	5	R5	6	R7	3	R8	4	R7	3	R8	2	R2	4	R6	6	12	11	13	8	10	**24**
64	R2	3	R5	6	R7	3	R8	4	R7	3	R8	2	R2	4	R6	6	12	11	13	8	10	**22**
65	R1	5	R3	1	R6	8	R4	9	R2	5	R1	7	R3	2	R6	6	17	13	8	20	15	**31**
66	R2	3	R3	1	R6	8	R4	9	R2	5	R1	7	R3	2	R6	6	17	13	8	20	15	**29**
67	R1	5	R5	6	R6	8	R4	9	R2	5	R1	7	R3	2	R6	6	22	18	13	20	15	**33**
68	R2	3	R5	6	R6	8	R4	9	R2	5	R1	7	R3	2	R6	6	22	18	13	20	15	**31**
69	R1	5	R3	1	R7	3	R4	9	R2	5	R1	7	R3	2	R6	6	17	13	8	15	10	**28**
70	R2	3	R3	1	R7	3	R4	9	R2	5	R1	7	R3	2	R6	6	17	13	8	15	10	**26**
71	R1	5	R5	6	R7	3	R4	9	R2	5	R1	7	R3	2	R6	6	22	18	13	15	10	**33**
72	R2	3	R5	6	R7	3	R4	9	R2	5	R1	7	R3	2	R6	6	22	18	13	15	10	**31**
73	R1	5	R3	1	R6	8	R8	4	R2	5	R1	7	R3	2	R6	6	12	13	8	20	15	**31**
74	R2	3	R3	1	R6	8	R8	4	R2	5	R1	7	R3	2	R6	6	12	13	8	20	15	**29**
75	R1	5	R5	6	R6	8	R8	4	R2	5	R1	7	R3	2	R6	6	17	18	13	20	15	**31**
76	R2	3	R5	6	R6	8	R8	4	R2	5	R1	7	R3	2	R6	6	17	18	13	20	15	**29**
77	R1	5	R3	1	R7	3	R8	4	R2	5	R1	7	R3	2	R6	6	12	13	8	15	10	**26**
78	R2	3	R3	1	R7	3	R8	4	R2	5	R1	7	R3	2	R6	6	12	13	8	15	10	**24**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
79	R1	5	R5	6	R7	3	R8	4	R2	5	R1	7	R3	2	R6	6	17	18	13	15	10	**29**
80	R2	3	R5	6	R7	3	R8	4	R2	5	R1	7	R3	2	R6	6	17	18	13	15	10	**27**
81	R1	5	R3	1	R6	8	R4	9	R7	3	R1	7	R3	2	R6	6	17	11	6	18	13	**29**
82	R2	3	R3	1	R6	8	R4	9	R7	3	R1	7	R3	2	R6	6	17	11	6	18	13	**27**
83	R1	5	R5	6	R6	8	R4	9	R7	3	R1	7	R3	2	R6	6	22	16	11	18	13	**33**
84	R2	3	R5	6	R6	8	R4	9	R7	3	R1	7	R3	2	R6	6	22	16	11	18	13	**31**
85	R1	5	R3	1	R7	3	R4	9	R7	3	R1	7	R3	2	R6	6	17	11	6	13	8	**28**
86	R2	3	R3	1	R7	3	R4	9	R7	3	R1	7	R3	2	R6	6	17	11	6	13	8	**26**
87	R1	5	R5	6	R7	3	R4	9	R7	3	R1	7	R3	2	R6	6	22	16	11	13	8	**33**
88	R2	3	R5	6	R7	3	R4	9	R7	3	R1	7	R3	2	R6	6	22	16	11	13	8	**31**
89	R1	5	R3	1	R6	8	R8	4	R7	3	R1	7	R3	2	R6	6	12	11	6	18	13	**29**
90	R2	3	R3	1	R6	8	R8	4	R7	3	R1	7	R3	2	R6	6	12	11	6	18	13	**27**
91	R1	5	R5	6	R6	8	R8	4	R7	3	R1	7	R3	2	R6	6	17	16	11	18	13	**29**
92	R2	3	R5	6	R6	8	R8	4	R7	3	R1	7	R3	2	R6	6	17	16	11	18	13	**27**
93	R1	5	R3	1	R7	3	R8	4	R7	3	R1	7	R3	2	R6	6	12	11	6	13	8	**24**
94	R2	3	R3	1	R7	3	R8	4	R7	3	R1	7	R3	2	R6	6	12	11	6	13	8	**22**
95	R1	5	R5	6	R7	3	R8	4	R7	3	R1	7	R3	2	R6	6	17	16	11	13	8	**28**
96	R2	3	R5	6	R7	3	R8	4	R7	3	R1	7	R3	2	R6	6	17	16	11	13	8	**26**
97	R1	5	R3	1	R6	8	R4	9	R2	5	R8	2	R3	2	R6	6	12	8	8	15	15	**26**
98	R2	3	R3	1	R6	8	R4	9	R2	5	R8	2	R3	2	R6	6	12	8	8	15	15	**24**
99	R1	5	R5	6	R6	8	R4	9	R2	5	R8	2	R3	2	R6	6	17	13	13	15	15	**28**
100	R2	3	R5	6	R6	8	R4	9	R2	5	R8	2	R3	2	R6	6	17	13	13	15	15	**26**
101	R1	5	R3	1	R7	3	R4	9	R2	5	R8	2	R3	2	R6	6	12	8	8	10	10	**23**
102	R2	3	R3	1	R7	3	R4	9	R2	5	R8	2	R3	2	R6	6	12	8	8	10	10	**21**
103	R1	5	R5	6	R7	3	R4	9	R2	5	R8	2	R3	2	R6	6	17	13	13	10	10	**28**
104	R2	3	R5	6	R7	3	R4	9	R2	5	R8	2	R3	2	R6	6	17	13	13	10	10	**26**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
105	R1	5	R3	1	R6	8	R8	4	R2	5	R8	2	R3	2	R6	6	7	8	8	15	15	**26**
106	R2	3	R3	1	R6	8	R8	4	R2	5	R8	2	R3	2	R6	6	7	8	8	15	15	**24**
107	R1	5	R5	6	R6	8	R8	4	R2	5	R8	2	R3	2	R6	6	12	13	13	15	15	**26**
108	R2	3	R5	6	R6	8	R8	4	R2	5	R8	2	R3	2	R6	6	12	13	13	15	15	**24**
109	R1	5	R3	1	R7	3	R8	4	R2	5	R8	2	R3	2	R6	6	7	8	8	10	10	**21**
110	R2	3	R3	1	R7	3	R8	4	R2	5	R8	2	R3	2	R6	6	7	8	8	10	10	**19**
111	R1	5	R5	6	R7	3	R8	4	R2	5	R8	2	R3	2	R6	6	12	13	13	10	10	**24**
112	R2	3	R5	6	R7	3	R8	4	R2	5	R8	2	R3	2	R6	6	12	13	13	10	10	**22**
113	R1	5	R3	1	R6	8	R4	9	R7	3	R8	2	R3	2	R6	6	12	6	6	13	13	**24**
114	R2	3	R3	1	R6	8	R4	9	R7	3	R8	2	R3	2	R6	6	12	6	6	13	13	**22**
115	R1	5	R5	6	R6	8	R4	9	R7	3	R8	2	R3	2	R6	6	17	11	11	13	13	**28**
116	R2	3	R5	6	R6	8	R4	9	R7	3	R8	2	R3	2	R6	6	17	11	11	13	13	**26**
117	R1	5	R3	1	R7	3	R4	9	R7	3	R8	2	R3	2	R6	6	12	6	6	8	8	**23**
118	R2	3	R3	1	R7	3	R4	9	R7	3	R8	2	R3	2	R6	6	12	6	6	8	8	**21**
119	R1	5	R5	6	R7	3	R4	9	R7	3	R8	2	R3	2	R6	6	17	11	11	8	8	**28**
120	R2	3	R5	6	R7	3	R4	9	R7	3	R8	2	R3	2	R6	6	17	11	11	8	8	**26**
121	R1	5	R3	1	R6	8	R8	4	R7	3	R8	2	R3	2	R6	6	7	6	6	13	13	**24**
122	R2	3	R3	1	R6	8	R8	4	R7	3	R8	2	R3	2	R6	6	7	6	6	13	13	**22**
123	R1	5	R5	6	R6	8	R8	4	R7	3	R8	2	R3	2	R6	6	12	11	11	13	13	**24**
124	R2	3	R5	6	R6	8	R8	4	R7	3	R8	2	R3	2	R6	6	12	11	11	13	13	**22**
125	R1	5	R3	1	R7	3	R8	4	R7	3	R8	2	R3	2	R6	6	7	6	6	8	8	**19**
126	R2	3	R3	1	R7	3	R8	4	R7	3	R8	2	R3	2	R6	6	7	6	6	8	8	**17**
127	R1	5	R5	6	R7	3	R8	4	R7	3	R8	2	R3	2	R6	6	12	11	11	8	8	**23**
128	R2	3	R5	6	R7	3	R8	4	R7	3	R8	2	R3	2	R6	6	12	11	11	8	8	**21**
129	R1	5	R3	1	R6	8	R4	9	R2	5	R1	7	R2	4	R4	4	17	13	10	20	17	**29**
130	R2	3	R3	1	R6	8	R4	9	R2	5	R1	7	R2	4	R4	4	17	13	10	20	17	**27**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
131	R1	5	R5	6	R6	8	R4	9	R2	5	R1	7	R2	4	R4	4	22	18	15	20	17	**31**
132	R2	3	R5	6	R6	8	R4	9	R2	5	R1	7	R2	4	R4	4	22	18	15	20	17	**29**
133	R1	5	R3	1	R7	3	R4	9	R2	5	R1	7	R2	4	R4	4	17	13	10	15	12	**26**
134	R2	3	R3	1	R7	3	R4	9	R2	5	R1	7	R2	4	R4	4	17	13	10	15	12	**24**
135	R1	5	R5	6	R7	3	R4	9	R2	5	R1	7	R2	4	R4	4	22	18	15	15	12	**31**
136	R2	3	R5	6	R7	3	R4	9	R2	5	R1	7	R2	4	R4	4	22	18	15	15	12	**29**
137	R1	5	R3	1	R6	8	R8	4	R2	5	R1	7	R2	4	R4	4	12	13	10	20	17	**29**
138	R2	3	R3	1	R6	8	R8	4	R2	5	R1	7	R2	4	R4	4	12	13	10	20	17	**27**
139	R1	5	R5	6	R6	8	R8	4	R2	5	R1	7	R2	4	R4	4	17	18	15	20	17	**29**
140	R2	3	R5	6	R6	8	R8	4	R2	5	R1	7	R2	4	R4	4	17	18	15	20	17	**27**
141	R1	5	R3	1	R7	3	R8	4	R2	5	R1	7	R2	4	R4	4	12	13	10	15	12	**24**
142	R2	3	R3	1	R7	3	R8	4	R2	5	R1	7	R2	4	R4	4	12	13	10	15	12	**22**
143	R1	5	R5	6	R7	3	R8	4	R2	5	R1	7	R2	4	R4	4	17	18	15	15	12	**27**
144	R2	3	R5	6	R7	3	R8	4	R2	5	R1	7	R2	4	R4	4	17	18	15	15	12	**25**
145	R1	5	R3	1	R6	8	R4	9	R7	3	R1	7	R2	4	R4	4	17	11	8	18	15	**27**
146	R2	3	R3	1	R6	8	R4	9	R7	3	R1	7	R2	4	R4	4	17	11	8	18	15	**25**
147	R1	5	R5	6	R6	8	R4	9	R7	3	R1	7	R2	4	R4	4	22	16	13	18	15	**31**
148	R2	3	R5	6	R6	8	R4	9	R7	3	R1	7	R2	4	R4	4	22	16	13	18	15	**29**
149	R1	5	R3	1	R7	3	R4	9	R7	3	R1	7	R2	4	R4	4	17	11	8	13	10	**26**
150	R2	3	R3	1	R7	3	R4	9	R7	3	R1	7	R2	4	R4	4	17	11	8	13	10	**24**
151	R1	5	R5	6	R7	3	R4	9	R7	3	R1	7	R2	4	R4	4	22	16	13	13	10	**31**
152	R2	3	R5	6	R7	3	R4	9	R7	3	R1	7	R2	4	R4	4	22	16	13	13	10	**29**
153	R1	5	R3	1	R6	8	R8	4	R7	3	R1	7	R2	4	R4	4	12	11	8	18	15	**27**
154	R2	3	R3	1	R6	8	R8	4	R7	3	R1	7	R2	4	R4	4	12	11	8	18	15	**25**
155	R1	5	R5	6	R6	8	R8	4	R7	3	R1	7	R2	4	R4	4	17	16	13	18	15	**27**
156	R2	3	R5	6	R6	8	R8	4	R7	3	R1	7	R2	4	R4	4	17	16	13	18	15	**25**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
157	R1	5	R3	1	R7	3	R8	4	R7	3	R1	7	R2	4	R4	4	12	11	8	13	10	**22**
158	R2	3	R3	1	R7	3	R8	4	R7	3	R1	7	R2	4	R4	4	12	11	8	13	10	**20**
159	R1	5	R5	6	R7	3	R8	4	R7	3	R1	7	R2	4	R4	4	17	16	13	13	10	**26**
160	R2	3	R5	6	R7	3	R8	4	R7	3	R1	7	R2	4	R4	4	17	16	13	13	10	**24**
161	R1	5	R3	1	R6	8	R4	9	R2	5	R8	2	R2	4	R4	4	12	8	10	15	17	**26**
162	R2	3	R3	1	R6	8	R4	9	R2	5	R8	2	R2	4	R4	4	12	8	10	15	17	**24**
163	R1	5	R5	6	R6	8	R4	9	R2	5	R8	2	R2	4	R4	4	17	13	15	15	17	**26**
164	R2	3	R5	6	R6	8	R4	9	R2	5	R8	2	R2	4	R4	4	17	13	15	15	17	**24**
165	R1	5	R3	1	R7	3	R4	9	R2	5	R8	2	R2	4	R4	4	12	8	10	10	12	**21**
166	R2	3	R3	1	R7	3	R4	9	R2	5	R8	2	R2	4	R4	4	12	8	10	10	12	**19**
167	R1	5	R5	6	R7	3	R4	9	R2	5	R8	2	R2	4	R4	4	17	13	15	10	12	**26**
168	R2	3	R5	6	R7	3	R4	9	R2	5	R8	2	R2	4	R4	4	17	13	15	10	12	**24**
169	R1	5	R3	1	R6	8	R8	4	R2	5	R8	2	R2	4	R4	4	7	8	10	15	17	**26**
170	R2	3	R3	1	R6	8	R8	4	R2	5	R8	2	R2	4	R4	4	7	8	10	15	17	**24**
171	R1	5	R5	6	R6	8	R8	4	R2	5	R8	2	R2	4	R4	4	12	13	15	15	17	**26**
172	R2	3	R5	6	R6	8	R8	4	R2	5	R8	2	R2	4	R4	4	12	13	15	15	17	**24**
173	R1	5	R3	1	R7	3	R8	4	R2	5	R8	2	R2	4	R4	4	7	8	10	10	12	**21**
174	R2	3	R3	1	R7	3	R8	4	R2	5	R8	2	R2	4	R4	4	7	8	10	10	12	**19**
175	R1	5	R5	6	R7	3	R8	4	R2	5	R8	2	R2	4	R4	4	12	13	15	10	12	**24**
176	R2	3	R5	6	R7	3	R8	4	R2	5	R8	2	R2	4	R4	4	12	13	15	10	12	**22**
177	R1	5	R3	1	R6	8	R4	9	R7	3	R8	2	R2	4	R4	4	12	6	8	13	15	**24**
178	R2	3	R3	1	R6	8	R4	9	R7	3	R8	2	R2	4	R4	4	12	6	8	13	15	**22**
179	R1	5	R5	6	R6	8	R4	9	R7	3	R8	2	R2	4	R4	4	17	11	13	13	15	**26**
180	R2	3	R5	6	R6	8	R4	9	R7	3	R8	2	R2	4	R4	4	17	11	13	13	15	**24**
181	R1	5	R3	1	R7	3	R4	9	R7	3	R8	2	R2	4	R4	4	12	6	8	8	10	**21**
182	R2	3	R3	1	R7	3	R4	9	R7	3	R8	2	R2	4	R4	4	12	6	8	8	10	**19**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
183	R1	5	R5	6	R7	3	R4	9	R7	3	R8	2	R2	4	R4	4	17	11	13	8	10	**26**
184	R2	3	R5	6	R7	3	R4	9	R7	3	R8	2	R2	4	R4	4	17	11	13	8	10	**24**
185	R1	5	R3	1	R6	8	R8	4	R7	3	R8	2	R2	4	R4	4	7	6	8	13	15	**24**
186	R2	3	R3	1	R6	8	R8	4	R7	3	R8	2	R2	4	R4	4	7	6	8	13	15	**22**
187	R1	5	R5	6	R6	8	R8	4	R7	3	R8	2	R2	4	R4	4	12	11	13	13	15	**24**
188	R2	3	R5	6	R6	8	R8	4	R7	3	R8	2	R2	4	R4	4	12	11	13	13	15	**22**
189	R1	5	R3	1	R7	3	R8	4	R7	3	R8	2	R2	4	R4	4	7	6	8	8	10	**19**
190	R2	3	R3	1	R7	3	R8	4	R7	3	R8	2	R2	4	R4	4	7	6	8	8	10	**17**
191	R1	5	R5	6	R7	3	R8	4	R7	3	R8	2	R2	4	R4	4	12	11	13	8	10	**22**
192	R2	3	R5	6	R7	3	R8	4	R7	3	R8	2	R2	4	R4	4	12	11	13	8	10	**20**
193	R1	5	R3	1	R6	8	R4	9	R2	5	R1	7	R3	2	R4	4	17	13	8	20	15	**29**
194	R2	3	R3	1	R6	8	R4	9	R2	5	R1	7	R3	2	R4	4	17	13	8	20	15	**27**
195	R1	5	R5	6	R6	8	R4	9	R2	5	R1	7	R3	2	R4	4	22	18	13	20	15	**31**
196	R2	3	R5	6	R6	8	R4	9	R2	5	R1	7	R3	2	R4	4	22	18	13	20	15	**29**
197	R1	5	R3	1	R7	3	R4	9	R2	5	R1	7	R3	2	R4	4	17	13	8	15	10	**26**
198	R2	3	R3	1	R7	3	R4	9	R2	5	R1	7	R3	2	R4	4	17	13	8	15	10	**24**
199	R1	5	R5	6	R7	3	R4	9	R2	5	R1	7	R3	2	R4	4	22	18	13	15	10	**31**
200	R2	3	R5	6	R7	3	R4	9	R2	5	R1	7	R3	2	R4	4	22	18	13	15	10	**29**
201	R1	5	R3	1	R6	8	R8	4	R2	5	R1	7	R3	2	R4	4	12	13	8	20	15	**29**
202	R2	3	R3	1	R6	8	R8	4	R2	5	R1	7	R3	2	R4	4	12	13	8	20	15	**27**
203	R1	5	R5	6	R6	8	R8	4	R2	5	R1	7	R3	2	R4	4	17	18	13	20	15	**29**
204	R2	3	R5	6	R6	8	R8	4	R2	5	R1	7	R3	2	R4	4	17	18	13	20	15	**27**
205	R1	5	R3	1	R7	3	R8	4	R2	5	R1	7	R3	2	R4	4	12	13	8	15	10	**24**
206	R2	3	R3	1	R7	3	R8	4	R2	5	R1	7	R3	2	R4	4	12	13	8	15	10	**22**
207	R1	5	R5	6	R7	3	R8	4	R2	5	R1	7	R3	2	R4	4	17	18	13	15	10	**27**
208	R2	3	R5	6	R7	3	R8	4	R2	5	R1	7	R3	2	R4	4	17	18	13	15	10	**25**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
209	R1	5	R3	1	R6	8	R4	9	R7	3	R1	7	R3	2	R4	4	17	11	6	18	13	**27**
210	R2	3	R3	1	R6	8	R4	9	R7	3	R1	7	R3	2	R4	4	17	11	6	18	13	**25**
211	R1	5	R5	6	R6	8	R4	9	R7	3	R1	7	R3	2	R4	4	22	16	11	18	13	**31**
212	R2	3	R5	6	R6	8	R4	9	R7	3	R1	7	R3	2	R4	4	22	16	11	18	13	**29**
213	R1	5	R3	1	R7	3	R4	9	R7	3	R1	7	R3	2	R4	4	17	11	6	13	8	**26**
214	R2	3	R3	1	R7	3	R4	9	R7	3	R1	7	R3	2	R4	4	17	11	6	13	8	**24**
215	R1	5	R5	6	R7	3	R4	9	R7	3	R1	7	R3	2	R4	4	22	16	11	13	8	**31**
216	R2	3	R5	6	R7	3	R4	9	R7	3	R1	7	R3	2	R4	4	22	16	11	13	8	**29**
217	R1	5	R3	1	R6	8	R8	4	R7	3	R1	7	R3	2	R4	4	12	11	6	18	13	**27**
218	R2	3	R3	1	R6	8	R8	4	R7	3	R1	7	R3	2	R4	4	12	11	6	18	13	**25**
219	R1	5	R5	6	R6	8	R8	4	R7	3	R1	7	R3	2	R4	4	17	16	11	18	13	**27**
220	R2	3	R5	6	R6	8	R8	4	R7	3	R1	7	R3	2	R4	4	17	16	11	18	13	**25**
221	R1	5	R3	1	R7	3	R8	4	R7	3	R1	7	R3	2	R4	4	12	11	6	13	8	**22**
222	R2	3	R3	1	R7	3	R8	4	R7	3	R1	7	R3	2	R4	4	12	11	6	13	8	**20**
223	R1	5	R5	6	R7	3	R8	4	R7	3	R1	7	R3	2	R4	4	17	16	11	13	8	**26**
224	R2	3	R5	6	R7	3	R8	4	R7	3	R1	7	R3	2	R4	4	17	16	11	13	8	**24**
225	R1	5	R3	1	R6	8	R4	9	R2	5	R8	2	R3	2	R4	4	12	8	8	15	15	**24**
226	R2	3	R3	1	R6	8	R4	9	R2	5	R8	2	R3	2	R4	4	12	8	8	15	15	**22**
227	R1	5	R5	6	R6	8	R4	9	R2	5	R8	2	R3	2	R4	4	17	13	13	15	15	**26**
228	R2	3	R5	6	R6	8	R4	9	R2	5	R8	2	R3	2	R4	4	17	13	13	15	15	**24**
229	R1	5	R3	1	R7	3	R4	9	R2	5	R8	2	R3	2	R4	4	12	8	8	10	10	**21**
230	R2	3	R3	1	R7	3	R4	9	R2	5	R8	2	R3	2	R4	4	12	8	8	10	10	**19**
231	R1	5	R5	6	R7	3	R4	9	R2	5	R8	2	R3	2	R4	4	17	13	13	10	10	**26**
232	R2	3	R5	6	R7	3	R4	9	R2	5	R8	2	R3	2	R4	4	17	13	13	10	10	**24**
233	R1	5	R3	1	R6	8	R8	4	R2	5	R8	2	R3	2	R4	4	7	8	8	15	15	**24**
234	R2	3	R3	1	R6	8	R8	4	R2	5	R8	2	R3	2	R4	4	7	8	8	15	15	**22**

Appendix 2 – Table 2. Detailed information about the project completion times using different resource alternatives (continues…)

	Node1		Node2		Node3		Node4		Node5		Node6		Node7		Node8		Path Completions					
	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	Res.	Time	2+4+6	2+5+6	2+5+7	3+5+6	3+5+7	**Cmax**
235	R1	5	R5	6	R6	8	R8	4	R2	5	R8	2	R3	2	R4	4	12	13	13	15	15	**24**
236	R2	3	R5	6	R6	8	R8	4	R2	5	R8	2	R3	2	R4	4	12	13	13	15	15	**22**
237	R1	5	R3	1	R7	3	R8	4	R2	5	R8	2	R3	2	R4	4	7	8	8	10	10	**19**
238	R2	3	R3	1	R7	3	R8	4	R2	5	R8	2	R3	2	R4	4	7	8	8	10	10	**17**
239	R1	5	R5	6	R7	3	R8	4	R2	5	R8	2	R3	2	R4	4	12	13	13	10	10	**22**
240	R2	3	R5	6	R7	3	R8	4	R2	5	R8	2	R3	2	R4	4	12	13	13	10	10	**20**
241	R1	5	R3	1	R6	8	R4	9	R7	3	R8	2	R3	2	R4	4	12	6	6	13	13	**22**
242	R2	3	R3	1	R6	8	R4	9	R7	3	R8	2	R3	2	R4	4	12	6	6	13	13	**20**
243	R1	5	R5	6	R6	8	R4	9	R7	3	R8	2	R3	2	R4	4	17	11	11	13	13	**26**
244	R2	3	R5	6	R6	8	R4	9	R7	3	R8	2	R3	2	R4	4	17	11	11	13	13	**24**
245	R1	5	R3	1	R7	3	R4	9	R7	3	R8	2	R3	2	R4	4	12	6	6	8	8	**21**
246	R2	3	R3	1	R7	3	R4	9	R7	3	R8	2	R3	2	R4	4	12	6	6	8	8	**19**
247	R1	5	R5	6	R7	3	R4	9	R7	3	R8	2	R3	2	R4	4	17	11	11	8	8	**26**
248	R2	3	R5	6	R7	3	R4	9	R7	3	R8	2	R3	2	R4	4	17	11	11	8	8	**24**
249	R1	5	R3	1	R6	8	R8	4	R7	3	R8	2	R3	2	R4	4	7	6	6	13	13	**22**
250	R2	3	R3	1	R6	8	R8	4	R7	3	R8	2	R3	2	R4	4	7	6	6	13	13	**20**
251	R1	5	R5	6	R6	8	R8	4	R7	3	R8	2	R3	2	R4	4	12	11	11	13	13	**22**
252	R2	3	R5	6	R6	8	R8	4	R7	3	R8	2	R3	2	R4	4	12	11	11	13	13	**20**
253	R1	5	R3	1	R7	3	R8	4	R7	3	R8	2	R3	2	R4	4	7	6	6	8	8	**17**
254	R2	3	R3	1	R7	3	R8	4	R7	3	R8	2	R3	2	R4	4	7	6	6	8	8	**15**
255	R1	5	R5	6	R7	3	R8	4	R7	3	R8	2	R3	2	R4	4	12	11	11	8	8	**21**
256	R2	3	R5	6	R7	3	R8	4	R7	3	R8	2	R3	2	R4	4	12	11	11	8	8	**19**

Appendix 3 - An example of a PROGEN generated 80-job project

****

****

file with basedata : XXX.BAS

initial value random generator: 447978243

****

****

projects : 1

jobs (incl. supersource/sink): 82

horizon : 640

RESOURCES

- renewable : 5 R
- nonrenewable : 0 N
- doubly constrained : 0 D

****

****

PROJECT INFORMATION:

pronr.	#jobs	rel.date	duedate	tardcost	MPM-Time
1	80	0	59	51	59

****

****

PRECEDENCE RELATIONS:

jobnr.	#modes	#successors	successors
1	1	1	2
2	3	4	3 5 7 27
3	3	3	4 6 33
4	3	1	9
5	3	4	8 12 24 30
6	3	5	16 34 46 54 72
7	3	4	13 18 19 25
8	3	2	26 38
9	3	5	10 21 23 31 53
10	3	3	11 20 60

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

11	3	2	14 28
12	3	3	17 37 64
13	3	2	33 80
14	3	2	15 35
15	3	1	62
16	3	2	39 51
17	3	1	45
18	3	2	55 78
19	3	1	49
20	3	1	22
21	3	3	40 42 56
22	3	1	43
23	3	1	67
24	3	2	41 54
25	3	1	44
26	3	2	29 35
27	3	1	69
28	3	1	78
29	3	2	36 74
30	3	1	71
31	3	2	32 58
32	3	1	74
33	3	1	75
34	3	2	59 61
35	3	1	81
36	3	1	66
37	3	1	47
38	3	1	48
39	3	1	76
40	3	1	52
41	3	1	44
42	3	2	50 68

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

43	3	1	51
44	3	1	63
45	3	2	60 72
46	3	1	70
47	3	1	76
48	3	2	57 62
49	3	1	78
50	3	1	79
51	3	1	73
52	3	1	71
53	3	1	57
54	3	1	63
55	3	1	57
56	3	1	65
57	3	2	60 66
58	3	1	79
59	3	1	68
60	3	1	75
61	3	1	75
62	3	1	80
63	3	1	64
64	3	1	77
65	3	1	79
66	3	2	72 75
67	3	1	70
68	3	1	69
69	3	1	75
70	3	1	80
71	3	1	76
72	3	2	77 80
73	3	1	77
74	3	1	76

Appendix 3 - An example of a PROGEN generated 80-job project (continues…)

75	3	1	81
76	3	1	77
77	3	1	79
78	3	1	81
79	3	1	81
80	3	1	81
81	3	1	82
82	1	0	

****

****

REQUESTS/DURATIONS:

jobnr.	mode	duration	R 1	R 2	R 3	R 4	R 5
1	1	0	0	0	0	0	0
2	1	1	1	0	0	0	0
	2	3	0	0	0	1	0
	3	4	0	0	0	0	1
3	1	4	0	0	0	1	0
	2	8	0	0	0	0	1
	3	8	1	0	0	0	0
4	1	9	0	0	0	0	1
	2	10	0	0	0	1	0
	3	10	0	0	1	0	0
5	1	1	1	0	0	0	0
	2	2	0	0	0	0	1
	3	4	0	0	1	0	0
6	1	2	0	0	0	1	0
	2	9	0	0	0	0	1
	3	10	0	1	0	0	0
7	1	6	0	1	0	0	0
	2	8	0	0	0	0	1
	3	10	0	0	0	1	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

8	1	5	0	0	0	0	1
	2	7	0	0	0	1	0
	3	10	0	1	0	0	0
9	1	1	0	1	0	0	0
	2	5	0	0	0	0	1
	3	10	0	0	1	0	0
10	1	9	0	0	0	0	1
	2	9	0	0	0	1	0
	3	10	1	0	0	0	0
11	1	2	0	0	1	0	0
	2	4	0	0	0	0	1
	3	4	0	1	0	0	0
12	1	1	0	0	0	1	0
	2	8	1	0	0	0	0
	3	8	0	1	0	0	0
13	1	1	1	0	0	0	0
	2	4	0	1	0	0	0
	3	8	0	0	1	0	0
14	1	1	0	0	0	0	1
	2	1	0	0	1	0	0
	3	5	0	1	0	0	0
15	1	1	1	0	0	0	0
	2	9	0	0	0	1	0
	3	10	0	1	0	0	0
16	1	5	0	0	0	0	1
	2	5	0	0	1	0	0
	3	9	0	0	0	1	0
17	1	2	0	0	0	1	0
	2	5	0	1	0	0	0
	3	9	0	0	1	0	0
18	1	1	0	0	0	0	1
	2	1	1	0	0	0	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

	3	6	0	1	0	0	0
19	1	1	0	0	1	0	0
	2	5	1	0	0	0	0
	3	7	0	0	0	1	0
20	1	7	1	0	0	0	0
	2	7	0	0	0	1	0
	3	7	0	1	0	0	0
21	1	3	0	1	0	0	0
	2	3	0	0	0	1	0
	3	6	1	0	0	0	0
22	1	1	0	0	0	1	0
	2	6	0	0	1	0	0
	3	6	0	1	0	0	0
23	1	1	0	0	0	1	0
	2	4	1	0	0	0	0
	3	7	0	0	0	0	1
24	1	6	1	0	0	0	0
	2	8	0	0	0	0	1
	3	10	0	0	0	1	0
25	1	1	0	0	1	0	0
	2	7	0	0	0	0	1
	3	10	0	1	0	0	0
26	1	3	1	0	0	0	0
	2	5	0	0	0	1	0
	3	8	0	0	0	0	1
27	1	9	0	1	0	0	0
	2	9	0	0	0	0	1
	3	9	1	0	0	0	0
28	1	1	0	1	0	0	0
	2	9	1	0	0	0	0
	3	10	0	0	0	0	1
29	1	1	0	0	0	0	1

Appendix 3 - An example of a PROGEN generated 80-job project (continues…)

	2	2	0	0	1	0	0
	3	9	0	0	0	1	0
30	1	2	0	1	0	0	0
	2	7	0	0	1	0	0
	3	10	1	0	0	0	0
31	1	1	0	0	1	0	0
	2	2	0	1	0	0	0
	3	6	0	0	0	0	1
32	1	4	0	0	1	0	0
	2	7	0	1	0	0	0
	3	9	0	0	0	0	1
33	1	5	1	0	0	0	0
	2	8	0	0	0	1	0
	3	9	0	1	0	0	0
34	1	3	0	0	0	1	0
	2	7	0	0	0	0	1
	3	7	0	0	1	0	0
35	1	3	1	0	0	0	0
	2	4	0	0	1	0	0
	3	5	0	1	0	0	0
36	1	3	0	0	0	1	0
	2	6	1	0	0	0	0
	3	8	0	0	0	0	1
37	1	5	0	0	1	0	0
	2	5	0	0	0	0	1
	3	7	0	0	0	1	0
38	1	1	1	0	0	0	0
	2	2	0	1	0	0	0
	3	6	0	0	0	1	0
39	1	1	1	0	0	0	0
	2	3	0	0	1	0	0
	3	9	0	0	0	1	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

40	1	1	1	0	0	0	0
	2	5	0	0	0	1	0
	3	7	0	0	0	0	1
41	1	9	0	0	1	0	0
	2	9	0	0	0	0	1
	3	10	0	1	0	0	0
42	1	1	0	0	1	0	0
	2	5	1	0	0	0	0
	3	8	0	0	0	1	0
43	1	3	1	0	0	0	0
	2	4	0	0	1	0	0
	3	8	0	0	0	0	1
44	1	2	0	1	0	0	0
	2	3	1	0	0	0	0
	3	5	0	0	1	0	0
45	1	2	0	0	0	1	0
	2	2	0	0	1	0	0
	3	3	1	0	0	0	0
46	1	9	0	1	0	0	0
	2	10	0	0	0	1	0
	3	10	0	0	0	0	1
47	1	2	1	0	0	0	0
	2	2	0	0	0	1	0
	3	8	0	1	0	0	0
48	1	1	0	0	0	1	0
	2	7	0	0	1	0	0
	3	10	0	1	0	0	0
49	1	1	0	1	0	0	0
	2	7	1	0	0	0	0
	3	9	0	0	1	0	0
50	1	4	0	0	1	0	0
	2	8	0	0	0	1	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues…)

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

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

	2	7	0	0	0	1	0
	3	9	0	0	0	0	1
62	1	4	0	0	0	0	1
	2	9	0	1	0	0	0
	3	9	0	0	0	1	0
63	1	5	0	0	0	1	0
	2	9	0	0	1	0	0
	3	10	0	0	0	0	1
64	1	1	0	1	0	0	0
	2	4	1	0	0	0	0
	3	8	0	0	0	1	0
65	1	2	0	0	0	1	0
	2	3	1	0	0	0	0
	3	8	0	0	0	0	1
66	1	1	0	0	0	0	1
	2	8	0	0	0	1	0
	3	9	1	0	0	0	0
67	1	1	1	0	0	0	0
	2	7	0	0	1	0	0
	3	10	0	1	0	0	0
68	1	2	0	1	0	0	0
	2	4	0	0	1	0	0
	3	10	0	0	0	1	0
69	1	3	0	0	1	0	0
	2	8	0	0	0	1	0
	3	9	0	0	0	0	1
70	1	4	0	0	0	0	1
	2	7	1	0	0	0	0
	3	9	0	0	1	0	0
71	1	2	0	0	0	1	0
	2	2	0	0	0	0	1
	3	7	0	1	0	0	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues…)

72	1	1	0	0	0	1	0
	2	1	0	1	0	0	0
	3	8	1	0	0	0	0
73	1	7	1	0	0	0	0
	2	8	0	0	0	0	1
	3	8	0	1	0	0	0
74	1	4	1	0	0	0	0
	2	5	0	0	1	0	0
	3	10	0	1	0	0	0
75	1	3	0	0	1	0	0
	2	4	1	0	0	0	0
	3	6	0	1	0	0	0
76	1	7	0	0	1	0	0
	2	8	0	0	0	0	1
	3	9	0	1	0	0	0
77	1	8	0	0	1	0	0
	2	10	1	0	0	0	0
	3	10	0	1	0	0	0
78	1	1	0	0	1	0	0
	2	3	0	1	0	0	0
	3	5	1	0	0	0	0
79	1	4	0	0	0	0	1
	2	6	0	1	0	0	0
	3	8	0	0	1	0	0
80	1	2	0	0	1	0	0
	2	5	0	0	0	0	1
	3	7	1	0	0	0	0
81	1	2	0	1	0	0	0
	2	2	0	0	1	0	0
	3	5	1	0	0	0	0
82	1	0	0	0	0	0	0

Appendix 3 - An example of a PROGEN generated 80-job project (continues...)

****

RESOURCEAVAILABILITIES:

R 1 R 2 R 3 R 4 R 5

1 1 1 1 1

****

Appendix 4 - Source File of a 80-job Project

82 5

1 1 1 1 1

0 0 0 0 0 1 2

1 0 0 3 4 4 3 5 7 27

8 0 0 4 8 3 4 6 33

0 0 10 10 9 1 9

1 0 4 0 2 4 8 12 24 30

0 10 0 2 9 5 16 34 46 54 72

0 6 0 10 8 4 13 18 19 25

0 10 0 7 5 2 26 38

0 1 10 0 5 5 10 21 23 31 53

10 0 0 9 9 3 11 20 60

0 4 2 0 4 2 14 28

8 8 0 1 0 3 17 37 64

1 4 8 0 0 2 33 80

0 5 1 0 1 2 15 35

1 10 0 9 0 1 62

0 0 5 9 5 2 39 51

0 5 9 2 0 1 45

1 6 0 0 1 2 55 78

5 0 1 7 0 1 49

7 7 0 7 0 1 22

6 3 0 3 0 3 40 42 56

0 6 6 1 0 1 43

4 0 0 1 7 1 67

6 0 0 10 8 2 41 54

0 10 1 0 7 1 44

3 0 0 5 8 2 29 35

9 9 0 0 9 1 69

9 1 0 0 10 1 78

0 0 2 9 1 2 36 74

10 2 7 0 0 1 71

Appendix 4 - Source File of a 80-job Project (continues…)

0 2 1 0 6 2 32 58

0 7 4 0 9 1 74

5 9 0 8 0 1 75

0 0 7 3 7 2 59 61

3 5 4 0 0 1 81

6 0 0 3 8 1 66

0 0 5 7 5 1 47

1 2 0 6 0 1 48

1 0 3 9 0 1 76

1 0 0 5 7 1 52

0 10 9 0 9 1 44

5 0 1 8 0 2 50 68

3 0 4 0 8 1 51

3 2 5 0 0 1 63

3 0 2 2 0 2 60 72

0 9 0 10 10 1 70

2 8 0 2 0 1 76

0 10 7 1 0 2 57 62

7 1 9 0 0 1 78

10 0 4 8 0 1 79

0 3 4 0 4 1 73

4 0 5 0 9 1 71

5 0 7 7 0 1 57

3 8 0 4 0 1 63

5 0 2 6 0 1 57

0 5 0 8 7 1 65

0 1 0 3 10 2 60 66

0 0 4 1 9 1 79

3 7 0 0 1 1 68

4 0 0 10 6 1 75

3 0 0 7 9 1 75

0 9 0 9 4 1 80

Appendix 4 - Source File of a 80-job Project (continues…)

0 0 9 5 10 1 64

4 1 0 8 0 1 77

3 0 0 2 8 1 79

9 0 0 8 1 2 72 75

1 10 7 0 0 1 70

0 2 4 10 0 1 69

0 0 3 8 9 1 75

7 0 9 0 4 1 80

0 7 0 2 2 1 76

8 1 0 1 0 2 77 80

7 8 0 0 8 1 77

4 10 5 0 0 1 76

4 6 3 0 0 1 81

0 9 7 0 8 1 77

10 10 8 0 0 1 79

5 3 1 0 0 1 81

0 6 8 0 4 1 81

7 0 2 0 5 1 81

5 2 2 0 0 1 82

0 0 0 0 0 0

0

CURRICULUM VITAE

Ayşe Ayçim (Atalay) Selam was born on july 29, 1978 in Istanbul, Turkey. After completing her secondary and high-school education in Beyoğlu Anatolian High School, she received her bachelor's (B.S.) degree from the Mechanical Engineering Department of Istanbul Technical University (ITU) in 2000 and her master's degree from the Industrial Engineering Department of Istanbul Technical University in Engineering Management program in 2002. Currently, she is a PhD. candidate at the Industrial Engineering Department of Marmara University.

She worked as a Mechanical Engineer in Motosan A.Ş. between 2001-2002, as a Planning Engineer at Karel Kalip between 2002 – 2006 and finally as a Research/Teaching Assistant at the Industrial Engineering Department of Marmara University between 2006 and 2011. Since 2010, she has been working as a Lecturer at the same department. Her research interests include project scheduling, optimization, green manufacturing, and sustainable supply chain management. She has been married to Mesut Selam since 2004 and has a three-and-a-half year-old daughter, Melike Sedef Selam.